



ESCUELA SUPERIOR DE INGENIERÍA

INGENIERÍA TÉCNICA EN INFORMÁTICA DE SISTEMAS

EDUTWITTER: MÓDULO PARA INTERACCIÓN ENTRE REDES SOCIALES Y SISTEMAS DE GESTIÓN DEL APRENDIZAJE

David Romero Santos

15 de noviembre de 2013



ESCUELA SUPERIOR DE INGENIERÍA

INGENIERÍA TÉCNICA EN INFORMÁTICA DE SISTEMAS

EDUTWITTER: MÓDULO PARA INTERACCIÓN ENTRE REDES SOCIALES Y
SISTEMAS DE GESTIÓN DEL APRENDIZAJE

- Departamento: Ingeniería Informática
- Directores del proyecto: Juan Manuel Dodero Beardo y Manuel Palomo Duarte
- Autor del proyecto: David Romero Santos

Cádiz, 15 de noviembre de 2013

Fdo: David Romero Santos

Agradecimientos

Quisiera dar las gracias a mi familia, especialmente a mis padres, que han permitido que pueda finalizar mis estudios y me han ayudado a convertirme en la persona que soy hoy. También a esos profesores y profesoras que durante todos estos años han contribuido de una forma u otra a mi formación, y no solo académicamente hablando. A mis tutores Manuel Palomo Duarte y Juan Manuel Dodero Beardo por transmitirme esas ganas y motivación de seguir adelante en cada reunión. A Daniel Cabeza y Juan Antonio Caballero del proyecto EVALfor por despejarme muchas de las dudas y orientarme en el desarrollo de bloques para Moodle al comienzo. Y por último, pero no menos importante, a los amigos y amigas que durante todo este tiempo se han preocupado y no han dejado de darme ánimos para terminar.

Resumen

EDUtwitter es un módulo cuya finalidad es crear una interacción bidireccional entre una red social y un sistema de gestión del aprendizaje (LMS).

En este caso se ha escogido la programación de un bloque para la plataforma Moodle, que es una aplicación web de tipo Ambiente Educativo Virtual de distribución libre usado en su campus virtual por la Universidad de Cádiz, entre otras. Como red social, se ha elegido trabajar con Twitter, que es un servicio de microblogging que nos permite realizar peticiones a su base de datos a través de una API pública (versión 1.1).

El objetivo es utilizar el bloque como recurso educativo, promoviendo entre los alumnos dar cortas opiniones sobre los cursos, avisándolos a través de sus cuentas personales de la red social de las novedades en los foros y demás actividades del campus virtual y recopilando información sobre la participación de los alumnos que pueden ser útiles para el docente en su evaluación.

Palabras clave: Red social LMS Moodle Twitter Bloque PHP Rest API

Índice general

I	Prolegómeno	1
1.	Introducción	5
1.1.	Motivación	5
1.2.	Alcance	6
1.2.1.	Usos de EDUtwitter en un curso	6
1.3.	Glosario de Términos	7
1.4.	Organización del documento	7
2.	Planificación	9
2.1.	Metodología de desarrollo	9
2.1.1.	Primera fase	9
2.1.2.	Primer ciclo	9
2.1.3.	Segundo ciclo	10
2.1.4.	Tercer ciclo	10
2.1.5.	Cuarto ciclo	10
2.2.	Planificación del proyecto	10
2.2.1.	Planificación prevista	10
2.2.2.	Planificación real	12
2.3.	Organización	12
2.4.	Costes	14
2.4.1.	Costes planificados	14
2.4.2.	Costes reales	14
2.5.	Riesgos	15
II	Desarrollo	17
3.	Análisis del Sistema	21
3.1.	Toma de requisitos	21
3.1.1.	Requisitos de interfaces externas	21
3.1.2.	Requisitos funcionales	21
3.1.3.	Requisitos de rendimiento	22
3.1.4.	Requisitos del sistema software	22
3.2.	Modelo Conceptual	22
3.2.1.	Diagrama de clases conceptuales	22
3.3.	Modelo de Casos de Uso	25
3.3.1.	Actores	25

3.3.2.	Diagrama de casos de uso	25
3.3.3.	Descripción de los casos de uso	25
3.4.	Modelo de Comportamiento	30
3.4.1.	Configurar bloque	30
3.4.2.	Enviar Tweet	30
3.4.3.	Generar Estadísticas	31
3.4.4.	Publicar Actividad Reciente	34
3.4.5.	Registrar Nombre Usuario	34
3.4.6.	Ver Tweets Recientes	34
3.5.	Modelo de Interfaz de Usuario	36
3.5.1.	Prototipos de interfaz gráfica	36
4.	Diseño del Sistema	43
4.1.	Arquitectura del Sistema	43
4.1.1.	Arquitectura Física	43
4.1.2.	Arquitectura Lógica	45
4.2.	Diseño Físico de Datos	47
4.2.1.	mdl_block_edutwitter_users	47
4.2.2.	mdl_block_edutwitter_params	47
4.2.3.	mdl_block_edutwitter_adds	49
4.3.	Análisis de las principales clases de la aplicación	49
4.4.	Diseño detallado de la Interfaz de Usuario	49
4.4.1.	Diagrama de interacción	50
4.4.2.	Interfaz	50
5.	Construcción del Sistema	53
5.1.	Entorno de Construcción	53
5.1.1.	Nivel de presentación	53
5.1.2.	Nivel de aplicación	53
5.1.3.	Nivel de persistencia	54
5.1.4.	Herramientas	54
5.2.	Código Fuente	54
5.3.	Scripts de Base de datos	56
5.3.1.	install.xml	56
6.	Pruebas del Sistema	59
6.1.	Estrategia	59
6.2.	Entorno de Pruebas	59
6.2.1.	Requisitos hardware	59
6.2.2.	Requisitos software	59
6.3.	Roles	60
6.4.	Niveles de Pruebas	60
6.4.1.	Pruebas Unitarias	60
6.4.2.	Pruebas de Integración	60
6.4.3.	Pruebas de Sistema	60
6.4.4.	Pruebas de Aceptación	60
6.5.	Resolución de errores	61
6.5.1.	Exceder el número de caracteres en un tweet	61

6.5.2. Publicación de recursos añadidos de forma oculta	61
6.5.3. Búsqueda exacta en Twitter	62
III Epílogo	63
7. Conclusiones	67
7.1. Objetivos alcanzados	67
7.2. Lecciones aprendidas	68
7.3. Trabajo futuro	69
A. Manual de implantación y explotación	71
A.1. Introducción	71
A.2. Requisitos previos	71
A.2.1. Requisitos hardware	71
A.2.2. Requisitos software	72
A.3. Inventario de componentes	72
A.3.1. Twitter-API-PHP	72
A.3.2. Twitter Search Block	72
A.4. Procedimientos de instalación	72
A.5. Pruebas de implantación	77
A.6. Procedimientos de operación y nivel de servicio	79
B. Manual de usuario	81
B.1. Introducción	81
B.2. Características	81
B.3. Requisitos previos	82
B.3.1. Requisitos del cliente	82
B.3.2. Requisitos del servidor	82
B.4. Uso del sistema	82
B.4.1. Uso como estudiante	82
B.4.2. Uso como profesor	84
C. Manual del desarrollador	87
C.1. Introducción	87
C.2. Uso de EDUtwitter por parte de otros módulos	87
C.3. Ampliar EDUtwitter	89
C.3.1. Añadir nuevo idioma	89
C.3.2. Añadir nuevas funciones	90
Bibliografía	94
GNU Free Documentation License	95
1. APPLICABILITY AND DEFINITIONS	95
2. VERBATIM COPYING	96
3. COPYING IN QUANTITY	97
4. MODIFICATIONS	97
5. COMBINING DOCUMENTS	99
6. COLLECTIONS OF DOCUMENTS	99

7. AGGREGATION WITH INDEPENDENT WORKS	99
8. TRANSLATION	100
9. TERMINATION	100
10. FUTURE REVISIONS OF THIS LICENSE	100
11. RELICENSING	101
ADDENDUM: How to use this License for your documents	101

Índice de figuras

2.1. Diagrama de Gantt planeado	11
2.2. Diagrama de Gantt real	13
3.1. Diagrama conceptual de clases	24
3.2. Actores	26
3.3. Diagrama de casos de uso	27
3.4. Diagrama de secuencia Configurar Bloque	31
3.5. Diagrama de secuencia Enviar Tweet	32
3.6. Diagrama de secuencia Generar Estadísticas	33
3.7. Diagrama de secuencia Publicar Actividad Reciente	35
3.8. Diagrama de secuencia Registrar Nombre Usuario	36
3.9. Diagrama de secuencia Ver Tweets Recientes	37
3.10. Modelo de Navegación de la interfaz gráfica	38
3.11. Interfaz Gráfica Pantalla Principal	39
3.12. Interfaz Gráfica Configurar	40
3.13. Interfaz Gráfica Registrar Nombre Usuario	40
3.14. Interfaz Gráfica Enviar Tweet	41
3.15. Interfaz Gráfica Estadísticas	41
3.16. Interfaz Gráfica Tabla	42
4.1. Arquitectura LAMP	44
4.2. Arquitectura Moodle	44
4.3. Arquitectura Lógica	46
4.4. Diseño de la Base de Datos	48
4.5. Diagrama de navegabilidad	50
4.6. Bloque con el rol de estudiante	51
4.7. Bloque con el rol de profesor	51
5.1. Líneas de código a lo largo del tiempo	56
A.1. Entorno Moodle	75
A.2. Página de inicio de Twitter	75
A.3. Seleccionar “My applications”	76
A.4. Datos de la aplicación	76
A.5. Configuración de un bloque EDUtwitter	78
B.1. Bloque con el rol de estudiante	83
B.2. Formulario para enviar el nombre de usuario en Twitter	84
B.3. Bloque con el rol de profesor	85

B.4. Formulario para enviar un tweet desde Moodle 85

B.5. Formulario para generar estadísticas 86

B.6. Tabla generada con las estadísticas por alumno 86

Índice de cuadros

2.1. Planificación de tareas prevista	12
2.2. Planificación de tareas real	14

Parte I

Prolegómeno

Capítulo 1

Introducción

1.1. Motivación

Estamos viviendo un momento en el que las redes sociales de internet se encuentran cada vez más presentes en nuestro día a día. A través de las aplicaciones móviles, la conexión se hace prácticamente permanente. Una vez más, la educación parece ir un paso por detrás de las innovaciones y la aplicación de las redes sociales en ésta se empieza a realizar de forma muy paulatina. Los docentes son conscientes de que una información enviada a través de las redes sociales llega mucho más rápidamente a los alumnos que la información enviada al campus virtual. Además, publicar un mensaje en dicho campus requiere mucho más tiempo para un docente que publicarlo en una red social, en el caso de que el tiempo de llegada del mensaje sea un factor muy importante, como el aviso del retraso de una clase o tutoría o el cambio de aula donde tendrá lugar un examen.

Este proyecto pretende hacer un acercamiento entre un sistema de gestión del aprendizaje y una red social, con el objetivo principal de mejorar la comunicación entre docentes y alumnos, promoviendo la retroalimentación de opiniones y apoyando la evaluación. Para conseguirlo, nos decantamos por Moodle[1] como sistema de gestión del aprendizaje y por Twitter[2] como red social.

Moodle es un sistema publicado bajo licencia libre, que cuenta con una gran comunidad de desarrolladores colaboradores. Tiene una filosofía pedagógica propia y está muy abierto a las colaboraciones. Se puede contribuir de varias formas: trabajando en el código del propio sistema para nuevas versiones, creando temas gráficos, aportando plugins... Todo debe hacerse conforme a una estructura y a un estilo de escritura de código acordados. Además la “M” de Moodle viene de modular. Es decir, las contribuciones se realizan de una forma muy cómoda sin tener que conocer exactamente cómo está implementado el resto del sistema. Aportas una parte siguiendo unas instrucciones y se integra perfectamente en la plataforma. Este proyecto se basa en realizar un bloque para la aplicación web Moodle. Para cumplir con los objetivos buscados se podría implementar en forma de actividad o de bloque. Las actividades son de tipo colaborativo y priorizan en la relación entre personas, además tiene sentido hablar de varias instancias distintas sobre una misma actividad para tratar distintos temas. Los bloques, por su parte, se colocan en las columnas laterales y son elementos de apoyo al proceso enseñanza-aprendizaje. Aportan normalmente información útil sobre el curso. No tiene mucho sentido tener más de una misma instancia de un bloque porque lo habitual es que duplique la información en la pantalla. Por lo tanto, como en este proyecto no existe la interacción entre personas (a no ser que se haga a través de la red social), ni tiene sentido mantener varias instancias de éste, nos decantamos por

la opción de bloque que cumple con todos los requisitos.

Twitter es una red social basada en el microblogging. Es una de las redes sociales más populares actualmente, por no decir la más popular. Su éxito reside en su sencillez. Los usuarios únicamente pueden enviar mensajes de hasta 140 caracteres. En ellos pueden poner un texto, una imagen o un enlace a una página web. Se pueden contestar entre mensajes usando una nomenclatura y hacer eco de los mensajes de otras personas. Normalmente los perfiles son públicos, sin necesidad de tener una cuenta registrada en Twitter para verlos. Cada usuario puede seguir a los usuarios que quiera (esto es, que le aparezcan en su página principal sus publicaciones) y ser seguido por otros usuarios, por lo que la relación no tiene por qué ser recíproca. Por su sencillez y por su uso habitual y de fácil acceso, ha sido la red social elegida para el proyecto.

1.2. Alcance

Este proyecto pretende formar parte de uno mayor donde se estudie la forma de conectar las redes sociales más populares con los sistemas de gestión del aprendizaje. La idea es utilizar el atractivo que las redes sociales tienen en los alumnos como apoyo a la docencia. Hacer interactuar entre sí las distintas API para recabar información de los distintos usos que se hacen de las redes sociales, hacia un enfoque educativo.

El objetivo general es conseguir relacionar o vincular de alguna forma una cuenta de Twitter con un curso en Moodle, al que además podremos referirnos a través de un hashtag o etiqueta en los mensajes en Twitter. También se pretende crear una comunicación bidireccional. Es decir, enviar datos a Twitter desde el curso, como un informe de las novedades o cualquier mensaje rápido, y recabar datos de Twitter para usarlos en el curso, por ejemplo, para aportar una opinión o apoyar la evaluación. También pretende aumentar la eficacia y rapidez en la comunicación entre docente y alumno.

No hay otro bloque parecido para Moodle, exceptuando uno que lo único que hace es mostrar tweets y que ya ha quedado inutilizado por el cambio de versión de la API de Twitter. El proyecto estará disponible en el repositorio oficial de bloques de Moodle, para que cualquier docente que use la plataforma pueda beneficiarse de su uso. Así, se le pretende dar visibilidad internacional, y por eso el proyecto está totalmente en inglés además de en castellano.

1.2.1. Usos de EDUtwitter en un curso

En este apartado se pretende mostrar el alcance que pueden tener las distintas funcionalidades implementadas en este bloque de la plataforma Moodle. Un alumno podrá comentar en tiempo real una opinión corta sobre el curso, sin tener que adecuarse a un formato de foro, simplemente con su teléfono móvil. También recibirá de forma más directa a través de la red social si alguien ha contestado o ha puesto un nuevo mensaje en el foro, si el profesor ha añadido nuevos recursos o actividades, si hay una tarea que entregar y qué compañeros la han entregado ya... Por otro lado, la figura del profesor también puede hacer uso de sus ventajas. El intercambio de opiniones cortas entre alumno le proporcionará retroalimentación o feedback de la materia que está impartiendo. Además podrá dar a los alumnos un aviso de forma rápida sin tener que adecuarse a la estructura formal que requiere un mensaje en Moodle. Así, podría advertir de forma muy sencilla y a través de su teléfono móvil a los alumnos sobre un retraso en la clase por cualquier percance, así como el cambio de aula para un examen. Estos son solo algunos ejemplos del potencial que tiene el hecho de enlazar un curso a una red social.

1.3. Glosario de Términos

- API: “Application Programming Interface”, en español Interfaz de “Programación de Aplicaciones” hace referencia al conjunto de procedimientos y funciones que ofrece una biblioteca para ser utilizado por otro software como una capa de abstracción.
- Bloque: Pequeña pantalla con información o herramientas que se puede mover alrededor de un curso de Moodle. Aporta nueva información sobre el curso, como listado de participantes, calendario, actividad reciente...
- E-learning: Se denomina aprendizaje electrónico a la educación a distancia completamente virtualizada a través de los nuevos canales electrónicos, utilizando para ello herramientas o aplicaciones de hipertexto como soporte de los procesos de enseñanza-aprendizaje.
- Hashtag: Una etiqueta o hashtag (del inglés hash, almohadilla o numeral y tag, etiqueta), es una cadena de caracteres formada por una o varias palabras concatenadas y precedidas por una almohadilla (#). Se usa en servicios web tales como Twitter, Facebook o en mensajería basada en protocolos IRC para señalar un tema sobre el que gira cierta conversación.
- LMS: “Learning Management System” o “Sistema de Gestión del Aprendizaje” es un software instalado en un servidor web que se emplea para administrar, distribuir y controlar las actividades de formación no presencial de una institución u organización. Las principales funciones del sistema de gestión de aprendizaje son: gestionar usuarios, recursos así como materiales y actividades de formación, administrar el acceso, controlar y hacer seguimiento del proceso de aprendizaje, realizar evaluaciones, generar informes, gestionar servicios de comunicación como foros de discusión, videoconferencias, entre otros.
- Microblogging: Servicio que permite a sus usuarios enviar y publicar mensajes breves, generalmente solo de texto.
- Moodle: Es una aplicación web de tipo LMS. De distribución libre, que ayuda a los educadores a crear comunidades de aprendizaje en línea. Estas herramientas son de gran utilidad en el ámbito educativo, ya que permiten a los profesores la gestión de cursos virtuales para sus alumnos, o la utilización de un espacio en línea que dé apoyo a la presencialidad.
- Tweet: Mensaje de texto plano de corta longitud, con un máximo de 140 caracteres que se envía a través de Twitter.
- Twitter: Red social que se basa en el servicio de microblogging. La red ha ganado popularidad mundialmente y se estima que tiene más de 200 millones de usuarios, generando 65 millones de tweets al día y maneja más de 800.000 peticiones de búsqueda diarias.

1.4. Organización del documento

En la primera parte nos encontramos con la planificación del proyecto y los costes estimados para llevarlo a cabo. En la segunda parte se describe el análisis y el diseño del sistema, es decir, todo el contenido relativo a la ingeniería del software. En la tercera y última parte podemos consultar el manual de instalación, del usuario y del desarrollador, las conclusiones y la bibliografía. Al final del documento se encuentra la información relativa a la licencia.

Capítulo 2

Planificación

En esta sección se describen todos los aspectos relativos a la gestión del proyecto: metodología, organización, costes, planificación y riesgos.

2.1. Metodología de desarrollo

Se ha elegido el modelo iterativo e incremental para el desarrollo del software. En una primera fase se realizó un estudio de las herramientas a utilizar. A partir de ahí comenzó la creación del software, se ha ido pasando por varios ciclos y en cada uno de ellos se ha añadido funciones al producto, creando varias versiones cada vez más completas. Las características o funcionalidades desarrolladas en cada versión las describimos a continuación.

2.1.1. Primera fase

Durante la primera fase el tiempo se dedicó a:

- Estudio del lenguaje de programación PHP. También se aprendió a escribir el código al estilo de programación dictado por Moodle.
- Estudio del funcionamiento de las API y en concreto un análisis de las capacidades de la API 1.0 de Twitter.
- Aprendizaje sobre la instalación y uso de un sistema Moodle en un servidor.

2.1.2. Primer ciclo

Una vez se asimilaron los conceptos básicos a manejar para el desarrollo del proyecto, se dieron los primeros pasos:

- Se realizó un estudio de los bloques disponibles para Moodle por si existía alguno que tuviera algunas funcionalidades parecidas.
- Se creó un primer bloque básico que ya era instalable en un sistema Moodle y tenía la estructura propia que mantendrá durante el resto del desarrollo.
- Se implementó la visualización de tweets recientes con un hashtag determinado en el bloque.

2.1.3. Segundo ciclo

En esta versión se completó todo lo referente a la estructura de Moodle, sin hacer ningún tipo de conexión más con Twitter:

- Modificación de la configuración básica del bloque, añadiendo todos los parámetros necesarios por las distintas funciones.
- Creación de la estructura necesaria para que el bloque pudiera ser utilizado en dos idiomas, inglés y castellano.
- Diseño e implementación de las tablas para la base de datos de Moodle a utilizar por el proyecto.
- Creación y procesamiento de los formularios que permiten realizar las funcionalidades del sistema.

2.1.4. Tercer ciclo

En este tercer ciclo el desarrollo se orientó a la interactividad del bloque de Moodle con la red social Twitter:

- Creación e implementación de las funciones necesarias para los tweets. Las funciones utilizadas para contar tweets, así como para acortarlos en caso necesario.
- Gestión de la utilización del script cron.php de Moodle por parte del bloque. Para poder tomar datos de Twitter y publicar la actividad reciente.
- Utilización de la librería para acceder a la API 1.0 de Twitter desde el bloque de Moodle.

2.1.5. Cuarto ciclo

En esta última versión se añadió la opción de publicar la actividad reciente en Twitter. Esto se hace a través del script cron.php de Moodle e incluye las novedades de alumnos, recursos y actividades añadidos, envío de tareas, mensajes en los foros y la posibilidad de enviar la actividad reciente de cualquier otro plugin que tenga implementada la función necesaria.

2.2. Planificación del proyecto

Procedemos ahora a comparar la planificación que se pensó hacer con la planificación real del desarrollo del proyecto a través de dos cuadros que detallan las tareas y sus respectivos diagramas de Gantt. Cada letra mayúscula se refiere a un ciclo del desarrollo y los números indican las diferentes tareas de dichos ciclos.

2.2.1. Planificación prevista

En el Cuadro 2.1 podemos ver una descripción detallada de las diferentes tareas planificadas en cada ciclo de desarrollo y la fechas en las que se programó que se iniciaran y se terminaran. En la Figura 2.1 podemos ver el Diagrama de Gantt general de esos mismos datos.

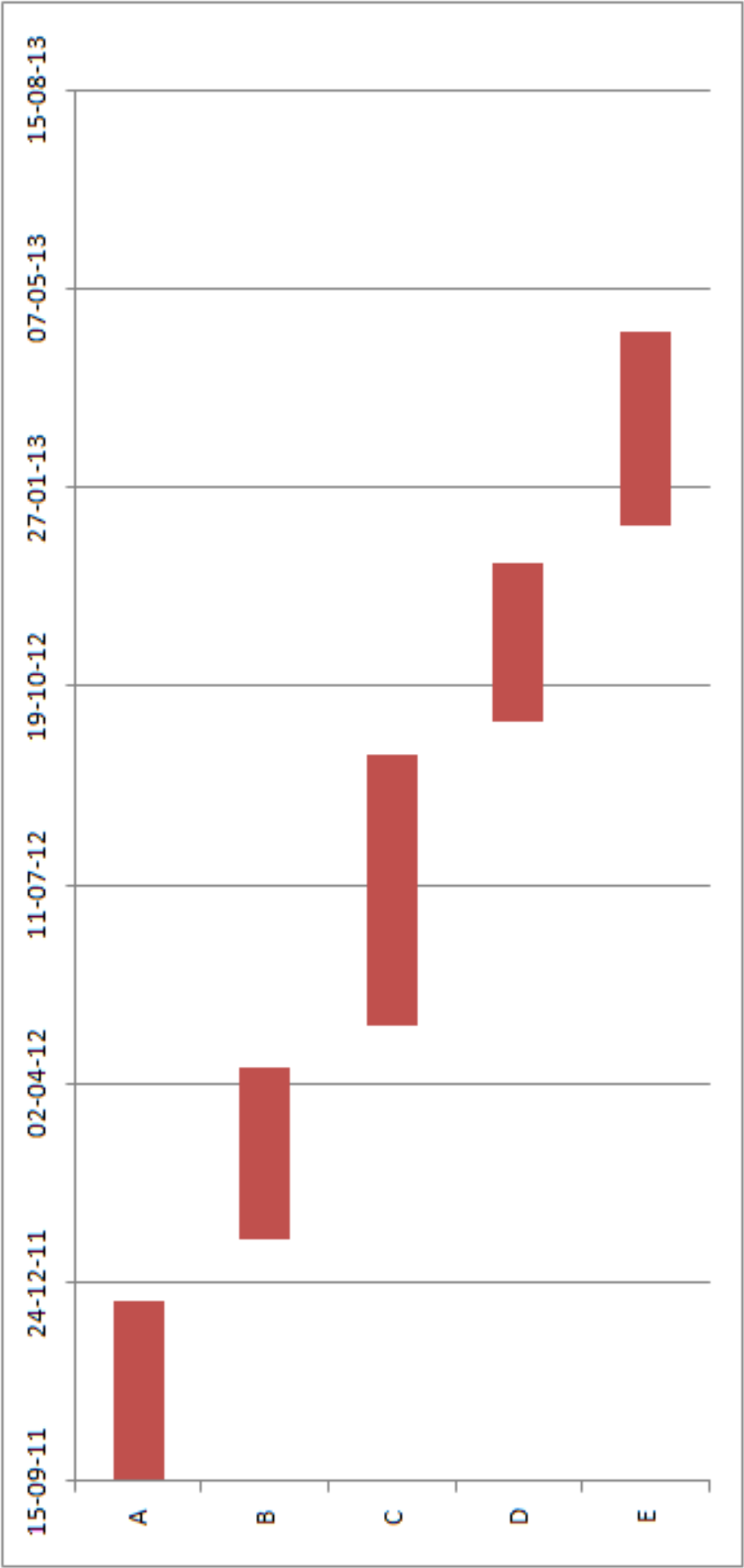


Figura 2.1: Diagrama de Gantt planeado

Tarea	Fecha de inicio	Fecha de fin	Descripción
A1	15/09/11		Estudio de PHP
A2			Estudio del API de Twitter
A3		15/12/11	Instalación y uso de Moodle
B1	15/01/12		Búsqueda de bloques similares
B2			Creación de bloque simple
B3		15/04/12	Visualización de tweets
C1	01/05/12		Configuración del bloque
C2			Gestión de idiomas
C3			Creación base de datos
C4		15/09/12	Creación de formularios
D1	01/10/12		Funciones para tweets
D2			Utilización de cron
D3		20/12/12	Acceder a Twitter desde PHP
E1	08/01/13	15/04/13	Publicar actividad reciente

Cuadro 2.1: Planificación de tareas prevista

2.2.2. Planificación real

En el Cuadro 2.2 podemos ver una descripción detallada de las diferentes tareas en cada ciclo de desarrollo y la fechas en las que se inició y se completó. En la Figura 2.2 podemos ver el Diagrama de Gantt detallado de esos mismos datos. La diferencia más sustancial con la planificación programada fue la inclusión de un nuevo ciclo forzado por la actualización de la API 1.0 de Twitter usada a la nueva versión que salió, la 1.1, que cambiaba totalmente la forma de acceder a los datos.

2.3. Organización

He sido el único desarrollador del proyecto, actuando como desarrollador y testeador al mismo tiempo conforme iban avanzando las versiones. Los tutores Juan Manuel Dodero y Manuel Palomo han sido los que me han guiado en los requisitos que buscaban que cumpliera el proyecto, actuando como clientes y evaluando una demostración del mismo. El hardware utilizado para el desarrollo ha sido el propio portátil del alumno, que ha servido de entorno de pruebas también haciendo uso del siguiente software:

- Ubuntu como sistema operativo.
- Apache como servidor local.
- Moodle como sistema de gestión del aprendizaje para las pruebas.
- MySQL como sistema de gestión de base de datos para almacenar los datos necesarios.
- API Rest de Twitter, primero 1.0 y luego 1.1.
- Librería Twitter-API-PHP para acceder al API 1.1 de Twitter a través de PHP.

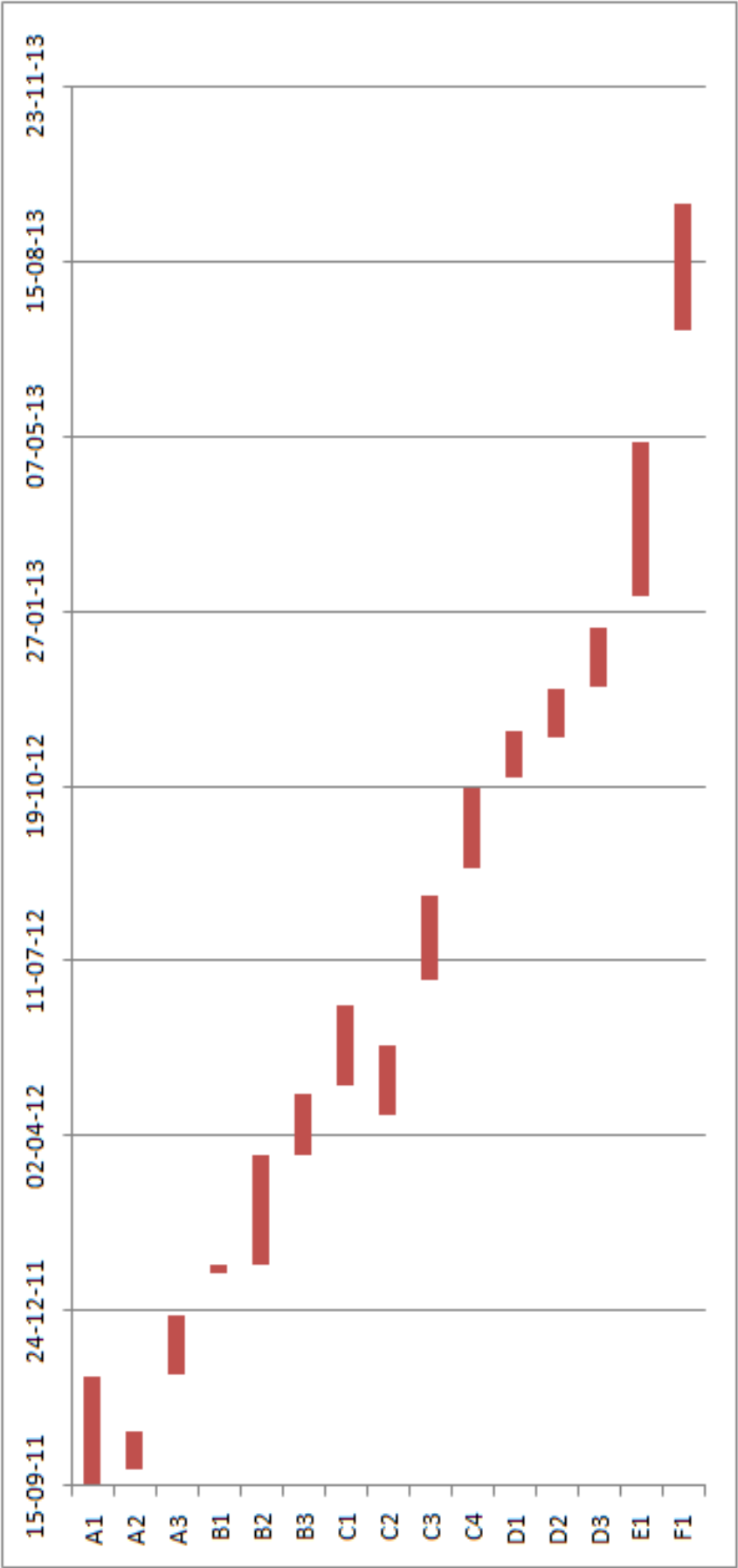


Figura 2.2: Diagrama de Gantt real

Tarea	Fecha de inicio	Fecha de fin	Descripción
A1	15/09/11		Estudio de PHP
A2			Estudio del API de Twitter
A3		22/12/11	Instalación y uso de Moodle
B1	15/01/12		Búsqueda de bloques similares
B2			Creación de bloque simple
B3		26/04/12	Visualización de tweets
C1	01/05/12		Configuración del bloque
C2			Gestión de idiomas
C3			Creación base de datos
C4		18/10/12	Creación de formularios
D1	24/10/12		Funciones para tweets
D2			Utilización de cron
D3		17/01/13	Acceder a Twitter desde PHP
E1	05/02/13	03/04/13	Actualizar todo a la API 1.1
F1	06/07/13	17/09/13	Publicar actividad reciente

Cuadro 2.2: Planificación de tareas real

- RedIRIS como forja para alojar el proyecto con Subversion como sistema de control de versiones.
- Blogger como sistema para alojar y publicar en el blog del proyecto.

2.4. Costes

No ha sido necesario ningún coste en material de hardware o software para la realización del proyecto. En lo referente al hardware, sólo ha sido necesario un portátil con una capacidad de procesamiento muy modesta, propiedad del desarrollador del proyecto. Las licencias de todo el software descrito en el apartado anterior y que ha sido utilizado son libres y por lo tanto no han requerido coste alguno. Procedemos ahora a hacer una comparativa del coste en materia de personal, es decir, recursos humanos, que hubiese supuesto este proyecto.

2.4.1. Costes planificados

Según el Diagrama de Gantt expuesto en la Figura 2.1, el proyecto se planificó para completarse en 496 días, aunque con una jornada de trabajo de 2 horas/día para poder compaginarlo con otros trabajos del desarrollador. En total serían 992 horas de trabajo o 6,2 personas/mes. Elegimos la categoría T.S. Apoyo a la Docencia e Investigación para calcular los costes según lo descrito en el Convenio[3]. El salario base mensual de cada trabajador junto con el complemento por la categoría hacen 2.746,65 €. Si calculamos el total estimado:

- 6,2 trabajadores x 2.746,65 € = **17.029,23 €**

2.4.2. Costes reales

Si ahora lo comparamos con el Diagrama de Gantt real de la Figura 2.2 vemos que el proyecto tardó en completarse 585 días. 89 días más de lo planificado, manteniendo la jornada laboral de 2

horas al día. Estos retrasos se deben a la compaginación del trabajo con otros estudios y trabajos eventuales. Recalculando, han sido necesarias 1170 horas de trabajo o 7,31 personas/mes. Por lo tanto, se ha producido una desviación en las personas/mes de 1.11. Si volvemos a calcular los costes según lo descrito en el apartado anterior, el total sería:

- 7,31 trabajadores x 2.746,65 € = **20.078,01 €**, que comparándolo con lo planificado resulta que se ha visto aumentado en 3048,78 €.

2.5. Riesgos

Podemos enumerar cuatro riesgos en EDUtwitter:

1. Caída del servicio de la API de Twitter.
2. Sobrepasar el límite de alguna función de la API de Twitter.
3. Cambios en el sistema Moodle.
4. Cambios en la API de Twitter.

Los dos primeros riesgos son bastante leves. Si se cayera el servidor que da acceso al API de Twitter, lo cual no he visto suceder nunca durante el desarrollo del proyecto, simplemente las funcionalidades que interactúan con ésta quedarían inútiles hasta que el servicio vuelva a ponerse en marcha. No hay nada que se pueda hacer para prevenirlo.

El envío de tweets no tiene límites, pero sí la búsqueda de éstos. En concreto se permiten 180 peticiones cada 15 minutos. Es un límite bastante amplio y es muy difícil llegar a éste, ya que los límites son independientes entre cursos porque pertenecen a cuentas distintas. En el caso de que se sobrepasara, simplemente el bloque no mostraría los tweets recientes durante los 15 minutos que se haya sobrepasado el límite.

Los dos siguientes riesgos pueden ser peligrosos o muy leves dependiendo de las diferencias que provoque el cambio. Si sacan una nueva versión de Moodle que maneje los bloques de forma totalmente distinta o cambie funciones usadas, EDUtwitter seguiría siendo compatible con las versiones anteriores del sistema. Además, se podría adaptar fácilmente a la nueva versión con un nuevo ciclo.

Algo parecido a esto último pasaría si se producen cambios en la API de Twitter. Aunque es bastante inusual, dado que desde su aparición sólo ha habido un cambio sustancial en la versión y ha sido este mismo año. Pero dado el caso de que cambiara la forma de acceso, habría que modificar cada fragmento de código donde se accede a ésta. Así que necesitaría de un nuevo ciclo de desarrollo para que el bloque tenga utilidad.

Parte II

Desarrollo

Capítulo 3

Análisis del Sistema

Esta sección cubre la toma de requisitos y el análisis del sistema de información a desarrollar, haciendo uso del lenguaje de modelado UML.

3.1. Toma de requisitos

Los requisitos se fueron definiendo entre los tutores del proyecto, que actuaban con el papel de clientes, mientras que el desarrollador se encargaba de estudiar si era posible cumplir esos requisitos con la limitación de la modularidad del sistema con el que se trabajaba. Finalmente, los requisitos definitivos quedan descritos en las siguientes secciones:

3.1.1. Requisitos de interfaces externas

Pasamos a describir los requisitos que definen las distintas pantallas con las que se puede encontrar el usuario:

- **Visión del bloque.** Esta es la visión general que se muestra en uno de los laterales de un curso. Aparecen los tweets recientes con el hashtag del curso. Dependiendo de si es un alumno o un profesor, verá al pie del bloque uno o tres enlaces a formularios respectivamente.
- **Configuración.** Sólo accesible por administradores del curso. En esta pantalla se podrán configurar todos los parámetros generales y específicos del bloque.
- **Enviar nombre de usuario.** Accesible por alumnos y profesores. Es un formulario utilizado para registrar en la base de datos el nombre de usuario en Twitter del usuario.
- **Enviar tweet.** Sólo accesible por profesores. A través de este formulario se podrá enviar un tweet a través de Twitter, añadiendo o no el hashtag propio del curso automáticamente.
- **Estadísticas.** Sólo accesible por profesores. Este formulario se utiliza para generar una tabla con las estadísticas de los alumnos, y reiniciar los contadores de la base de datos si se buscase ese objetivo.

3.1.2. Requisitos funcionales

Las funciones que se podrán realizar dependerá del rol del usuario:

■ Alumnos y profesores

- Ver los tweets recientes con el hashtag del curso.
- Enviar el nombre de usuario en Twitter para registrarlo en una base de datos junto con su nombre real, para poder generar las estadísticas de ese usuario.

■ Profesores

- Enviar tweet controlando el número de caracteres y con la posibilidad de añadir el hashtag del curso automáticamente.
- Generar estadísticas de los usuarios del bloque para apoyar la evaluación.

■ Administradores del curso

- Enlazar una cuenta de Twitter con el curso.
- Elegir un hashtag propio para el curso y configurarlo.
- Elegir el idioma en que los tweets automáticos van a publicarse en Twitter.
- Configurar la visualización de los tweets recientes: número de tweets, si mostrar nombres reales, si mostrar nombres de usuario y si mostrar imágenes de perfil.
- Elegir si se publica en Twitter la actividad reciente.
- Decidir si se pueden generar estadísticas.

3.1.3. Requisitos de rendimiento

Esta aplicación va a ejecutarse en un servidor donde esté instalado el sistema Moodle. Por lo tanto, debe adecuar su rendimiento y requisitos de hardware a los del propio Moodle, intentando que no realentice en absoluto a la plataforma y minimizando los recursos a utilizar. Las funciones de mantenimiento del mismo se realizarán a la vez que el resto del sistema a través del script proporcionado por la plataforma.

3.1.4. Requisitos del sistema software

Los requisitos del sistema son los que corresponden a un bloque para la plataforma Moodle. Debe tener la estructura y cumplir los estándares necesarios para que pueda instalarse en las versiones más recientes de este sistema. Además de cumplir los requisitos necesarios para ser admitido por la comunidad libre de Moodle.

3.2. Modelo Conceptual

A partir de los requisitos de información, desarrollamos un diagrama conceptual de clases UML en la Figura 3.1, identificando las clases, atributos, relaciones, restricciones adicionales y reglas de derivación necesarias.

3.2.1. Diagrama de clases conceptuales

Comienzo el diagrama partiendo de la entidad “Plataforma Moodle”, aunque realmente el proyecto desarrollado comienza en la entidad “EDUtwitter”, he aprovechado para ilustrar un poco mejor qué es lo que representa del sistema general.

Plataforma Moodle

Esta es la clase global del sistema para el que estamos desarrollando. La Plataforma Moodle contiene a todos los cursos.

Curso

Cada curso trata sobre una materia en concreto. En éstos, los profesores añaden los contenidos necesarios para el aprendizaje que son consultados y utilizados por los alumnos. Los elementos que se pueden añadir en un curso los dividimos en actividades, recursos y bloques.

Actividad, Recurso y Bloque

En las actividades los alumnos interactúan entre ellos o con el profesor. Los recursos son objetos que los profesores establecen para asistir el aprendizaje, como un archivo o un enlace. Por último, los bloques aportan información a la interfaz del curso. EDUtwitter es uno de ellos.

EDUtwitter

A partir de esta clase comienza la parte desarrollada para el proyecto. Esta es la clase general, una especialización de la clase Bloque. Sus funciones son mostrar los tweets recientes y enlazar con las otras funcionalidades (dependiendo del rol del usuario), además de encargarse de las tareas de mantenimiento.

Usuario

Esta clase gestiona y hace uso de todo el contenido de los cursos y, específicamente, del bloque EDUtwitter que es el que exponemos ahora. Dependiendo del rol (Administrador, Profesor o Estudiante) del usuario podrá realizar unas funciones u otras.

Administrador, Profesor y Estudiante

El Administrador puede añadir y configurar el bloque, además de hacer uso como un Profesor. El Profesor puede hacer uso de todas sus funcionalidades una vez que está instalado y configurado y el Alumno puede hacer un uso parcial del bloque.

Tweet

La entidad Tweet es necesaria para enviar un mensaje desde la plataforma Moodle a la red social Twitter. Pueden hacerlo los administradores y los profesores.

Tabla estadísticas

Las tablas estadísticas contienen la información relativa a los tweets de los alumnos y profesores. Pueden acceder a ella administradores y profesores.

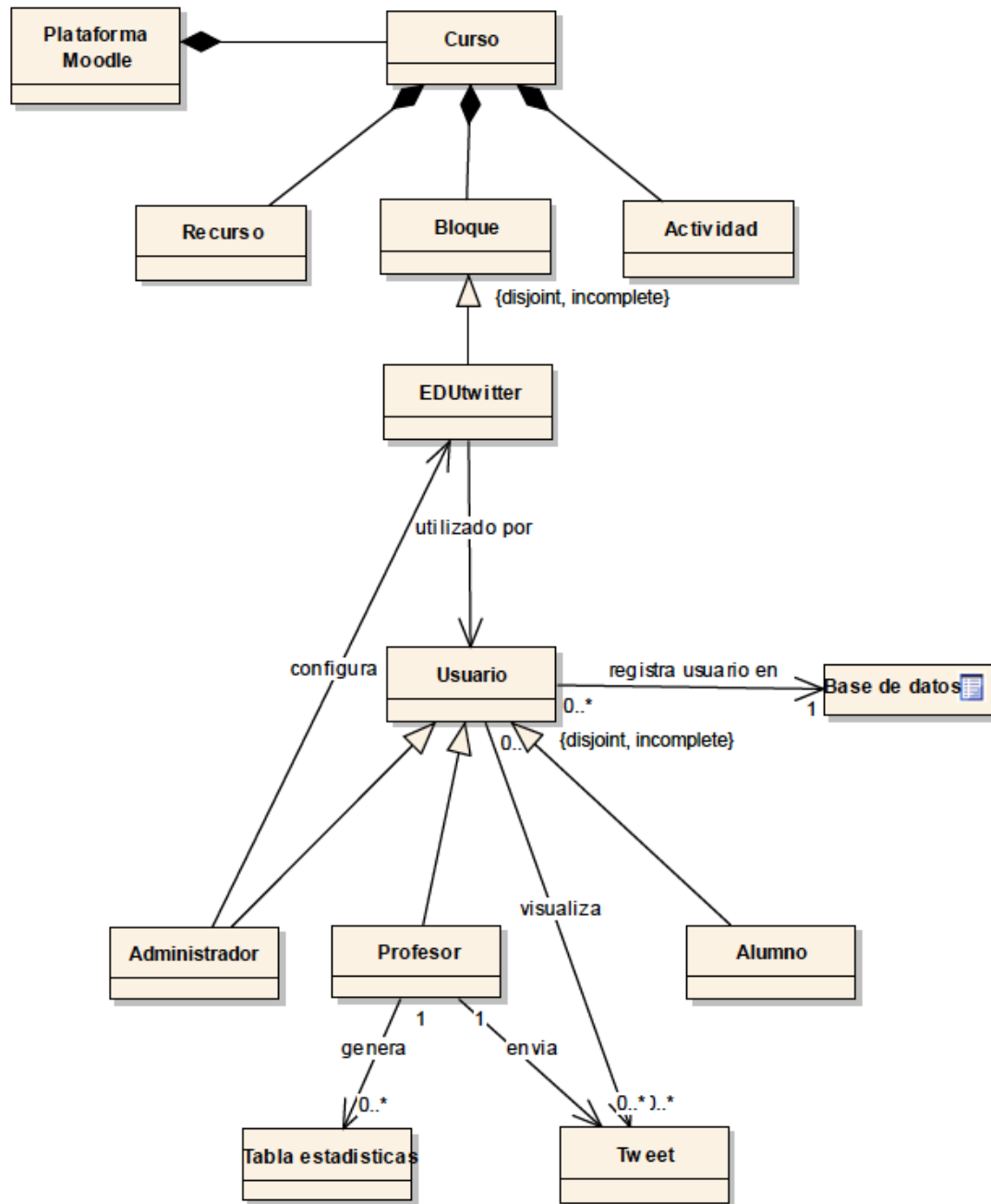


Figura 3.1: Diagrama conceptual de clases

3.3. Modelo de Casos de Uso

3.3.1. Actores

En este apartado se describirán los diferentes roles que juegan los usuarios que interactúan con el sistema. Los actores pueden ser roles de personas físicas, sistemas externos o incluso el tiempo (eventos temporales). En la Figura 3.2 podemos verlos en un diagrama. El Administrador Moodle es la persona física que gestiona en el servidor la plataforma Moodle, que se encargará del script de mantenimiento que es el que realiza la publicación de la actividad reciente. El Estudiante puede ver todo el contenido del curso y participar en las actividades, en lo referente al bloque EDUtwitter puede ver los tweets recientes y registrar su nombre de usuario. El Profesor además de eso puede enviar tweets y generar estadísticas. Y por último, el Administrador del curso puede hacer todo eso y además configurar el bloque. Por eso los he colocado como herencia uno de otro en el diagrama, ya que el Profesor hereda algunas funcionalidades del Administrador del curso y el Estudiante hace lo mismo con el Profesor.

3.3.2. Diagrama de casos de uso

En la Figura 3.3 se encuentra el diagrama de los casos de uso que vamos a describir en el apartado siguiente. Se ha tomado cada rol como un actor distinto para identificar mejor las funciones de cada uno.

3.3.3. Descripción de los casos de uso

Pasamos a describir ahora cada caso de uso reflejado en el diagrama.

Caso de uso: Configurar bloque

- Descripción: El usuario desea configurar los parámetros del bloque para controlar su comportamiento.
- Actores: Administrador del curso.
- Precondiciones: El bloque se encuentra correctamente instalado en el sistema y añadido en el curso.
- Postcondiciones: Se guardarán los parámetros necesarios y podrá comenzar el uso del bloque.
- Escenario principal:
 1. El usuario pulsará en el botón “Activar edición” en la página principal del curso.
 2. El sistema permitirá ahora la edición de los contenidos.
 3. El usuario pulsará el botón correspondiente a “Configuración” en la barra superior del bloque EDUtwitter.
 4. El sistema mostrará la pantalla de configuración del bloque.
 5. El usuario introducirá las claves de Twitter necesarias. También generará un hashtag, elegirá el idioma de publicación en Twitter, seleccionará el número de tweets recientes que se van a visualizar en el bloque, los parámetros de configuración de visualización, si desea que se publique la actividad reciente en Twitter y si quiere que se puedan generar estadísticas. Después, pulsará en el botón “Guardar cambios”.

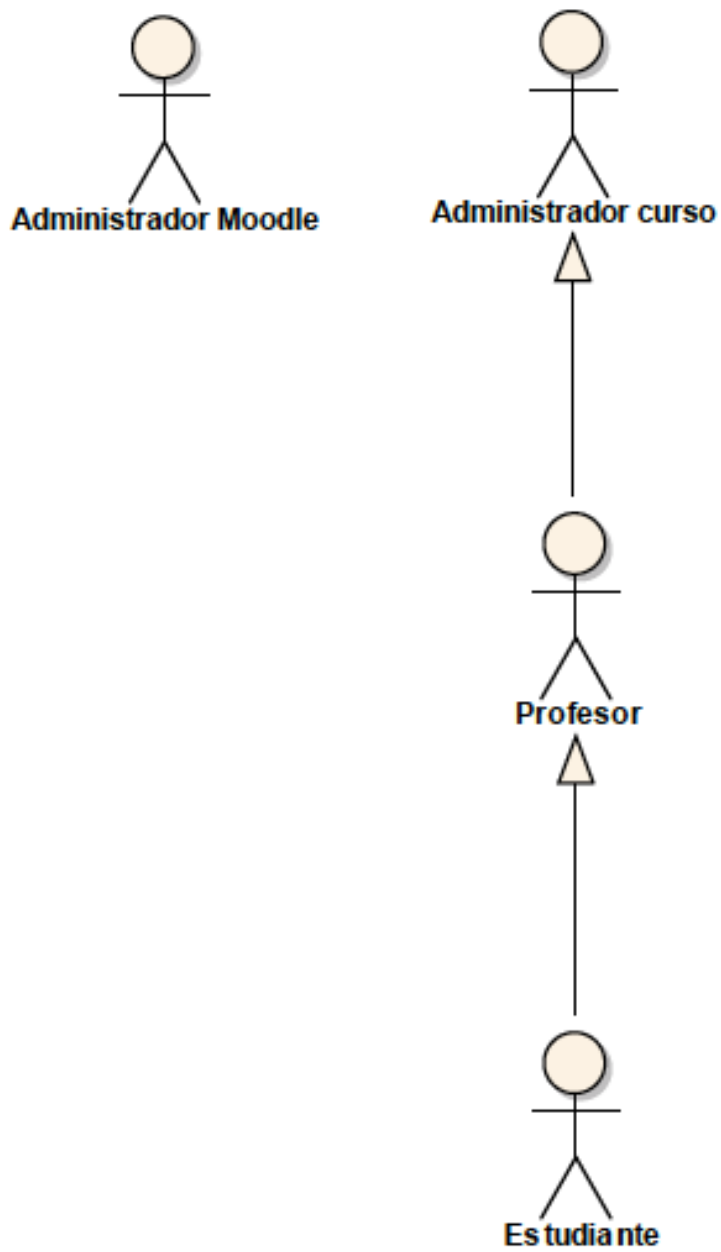


Figura 3.2: Actores

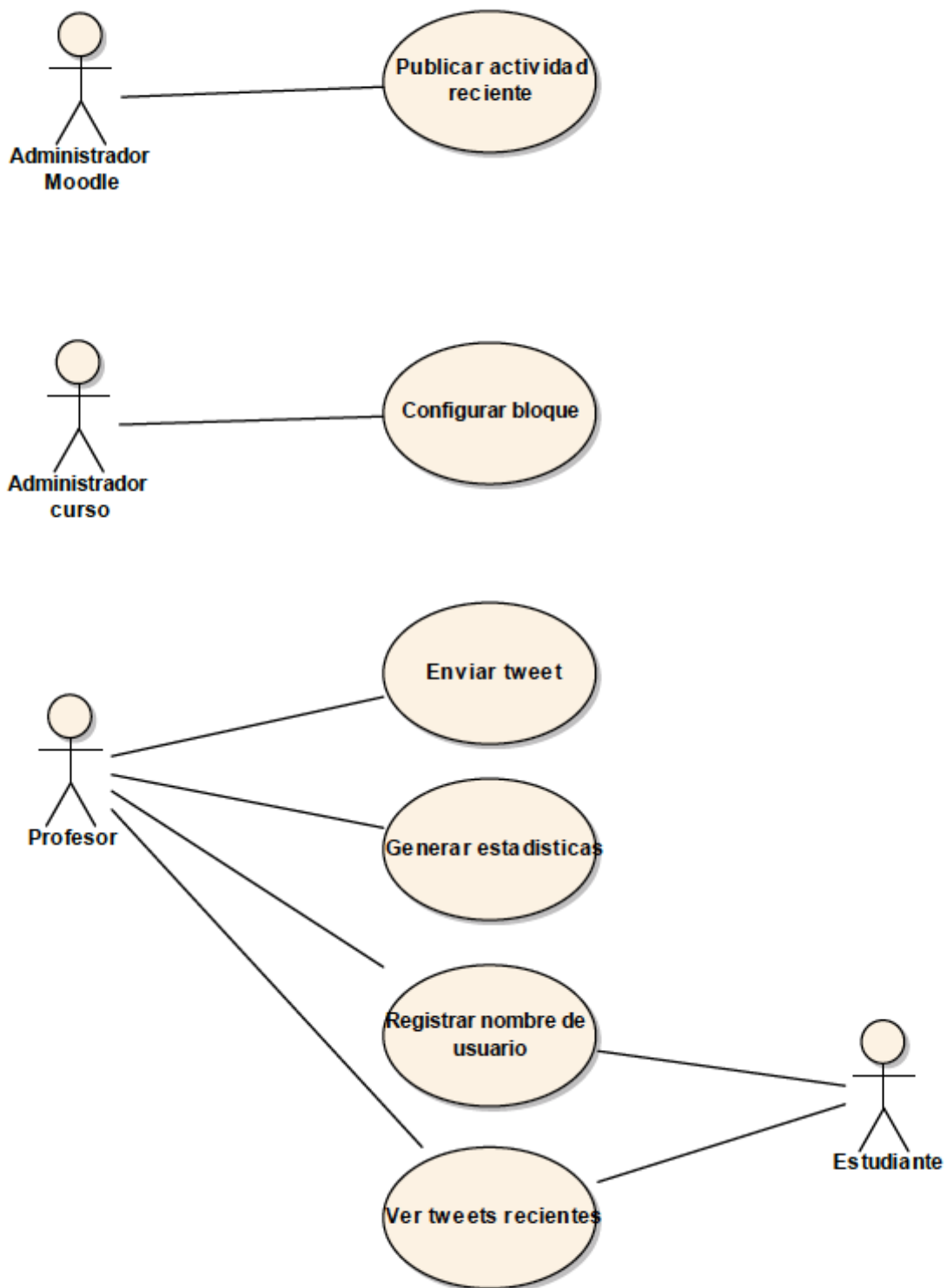


Figura 3.3: Diagrama de casos de uso

6. El sistema guardará los cambios y volverá a mostrar la pantalla principal del curso.

■ Extensiones - flujo alternativo:

- *a. El usuario pulsa el botón “Cancelar”, y vuelve a la pantalla principal del curso.

Caso de uso: Enviar tweet

■ Descripción: El usuario desea enviar un tweet desde la interfaz de Moodle.

■ Actores: Profesor.

■ Precondiciones: El bloque se encuentra correctamente instalado en el sistema, añadido en el curso y configurado.

■ Postcondiciones: Se enviará un tweet desde la cuenta de Twitter enlazada con el curso.

■ Escenario principal:

1. El usuario pulsa el enlace “Enviar tweet” en el pie de página del bloque.
2. El sistema mostrará un formulario.
3. El usuario elegirá si desea incluir el hashtag del curso automáticamente.
4. El usuario escribirá el tweet que desea enviar. Después, pulsará en el botón “Guardar cambios”.
5. El sistema enviará el tweet y volverá a mostrar la pantalla principal del curso.

■ Extensiones - flujo alternativo:

- *a. El usuario pulsa el botón “Cancelar”, y vuelve a la pantalla principal del curso.
- 5a. El tweet escrito por el usuario excede los 250 caracteres y el sistema exige que lo corrija antes de poder pulsar el botón “Guardar cambios”.

Caso de uso: Generar estadísticas

■ Descripción: El usuario desea generar estadísticas del uso de Twitter por parte de los usuarios del curso.

■ Actores: Profesor.

■ Precondiciones: El bloque se encuentra correctamente instalado en el sistema, añadido en el curso y configurado.

■ Postcondiciones: Se generará una tabla con las estadísticas de los usuarios del curso.

■ Escenario principal:

1. El usuario pulsa el enlace “Estadísticas” en el pie de página del bloque.
2. El sistema mostrará un formulario.
3. El usuario pulsará en el botón “Guardar cambios”.
4. El sistema mostrará una tabla con las estadísticas.

- Extensiones - flujo alternativo:
 - *a. El usuario pulsa el botón “Cancelar”, y vuelve a la pantalla principal del curso.
 - 3a. El usuario marca la casilla “Reiniciar estadísticas” y luego pulsa el botón “Guardar cambios”. El sistema mostrará una tabla con las estadísticas y reiniciará los contadores usados para generar las estadísticas.

Caso de uso: Registrar nombre de usuario

- Descripción: El usuario desea registrar su nombre de usuario en Twitter.
- Actores: Profesor o Estudiante.
- Precondiciones: El bloque se encuentra correctamente instalado en el sistema y añadido en el curso.
- Postcondiciones: El nombre de usuario en Twitter quedará registrado junto con el nombre real del usuario.
- Escenario principal:
 1. El usuario pulsa el enlace “Enviar usuario en Twitter” en el pie de página del bloque.
 2. El sistema mostrará un formulario.
 3. El usuario escribirá su nombre de usuario en Twitter y pulsará en el botón “Guardar cambios”.
 4. El sistema mostrará la pantalla principal del curso.
- Extensiones - flujo alternativo:
 - *a. El usuario pulsa el botón “Cancelar”, y vuelve a la pantalla principal del curso.

Caso de uso: Ver tweets recientes

- Descripción: El usuario desea consultar los tweets recientes con el hashtag del curso.
- Actores: Profesor o Estudiante.
- Precondiciones: El bloque se encuentra correctamente instalado en el sistema, añadido en el curso y configurado.
- Postcondiciones: Ninguna.
- Escenario principal:
 1. El usuario busca en la página principal del curso el bloque EDUtwitter.
 2. El sistema mostrará en el bloque los tweets recientes con el hashtag del curso.

Caso de uso: Publicar actividad reciente

- Descripción: El usuario desea que se proceda a la publicación en Twitter de la actividad reciente del curso.
- Actores: Administrador Moodle.
- Precondiciones: El bloque se encuentra correctamente instalado en el sistema, añadido en el curso y configurado.
- Postcondiciones: Quedará reflejada la actividad reciente del curso en Twitter a través de los tweets que sean necesarios.
- Escenario principal:
 1. El usuario ejecuta el script de mantenimiento del sistema, cron.php.
 2. El sistema mostrará un log con los cambios realizados.

3.4. Modelo de Comportamiento

A partir de los casos de uso anteriores, se crea el modelo de comportamiento. Para ello, se realizarán los diagramas de secuencia del sistema, donde se identificarán las operaciones o servicios del sistema. Luego, se detallará la descripción de las operaciones identificadas.

3.4.1. Configurar bloque

El diagrama de secuencia correspondiente se puede ver en la Figura 3.4. Las operaciones utilizadas son:

- void Mostrar_Pprincipal(). El sistema muestra la página principal donde se encuentra el bloque al usuario.
- void Edicion(int). El usuario indica que quiere editar la configuración del bloque enviando al sistema el id correspondiente.
- void Mostrar_Pconfiguracion(). El sistema muestra la página de configuración del bloque con el id dado.
- void Configurar(array). El usuario envía al sistema un array con los parámetros de configuración que desea.
- void Guardar(array). El sistema almacena dichos parámetros de configuración a través de la base de datos.

3.4.2. Enviar Tweet

El diagrama de secuencia correspondiente se puede ver en la Figura 3.5. Las operaciones utilizadas son:

- void Mostrar_Pprincipal(). El sistema muestra la página principal donde se encuentra el bloque al usuario.

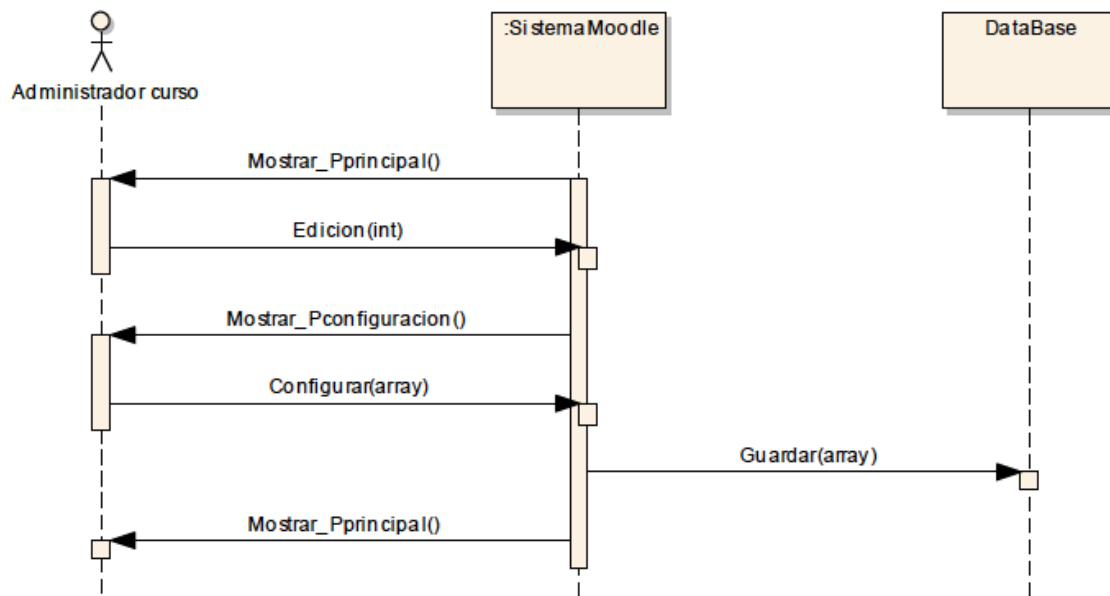


Figura 3.4: Diagrama de secuencia Configurar Bloque

- void FormularioEnvio(). EL usuario pide al sistema el formulario de envío de tweets.
- void Mostrar_Pformularioenvio(). El sistema muestra al usuario el formulario de envío de tweets.
- void ContenidoTweet(string,bool). El usuario pasa al sistema la cadena de caracteres que servirá como contenido del tweet y si desea incluir automáticamente el hashtag del curso o no.
- void PedirDatos(int). El sistema pide a la base de datos los parámetros correspondientes de la cuenta de Twitter asociada al bloque con el id dado.
- void DevolverDatos(array). La base de datos devuelve al sistema los datos pedidos.
- void EnviarTweet(string,array). El sistema envía a Twitter el contenido de un tweet con su correspondiente autenticación necesaria.
- void PublicarTweet(string,array). Twitter publica el tweet enviado por el sistema.
- void CodigoEnvio(int). Twitter envía al sistema un código de respuesta para que el sistema sepa que la publicación se ha efectuado correctamente.

3.4.3. Generar Estadísticas

El diagrama de secuencia correspondiente se puede ver en la Figura 3.6. Las operaciones utilizadas son:

- void Mostrar_Pprincipal(). El sistema muestra la página principal donde se encuentra el bloque al usuario.

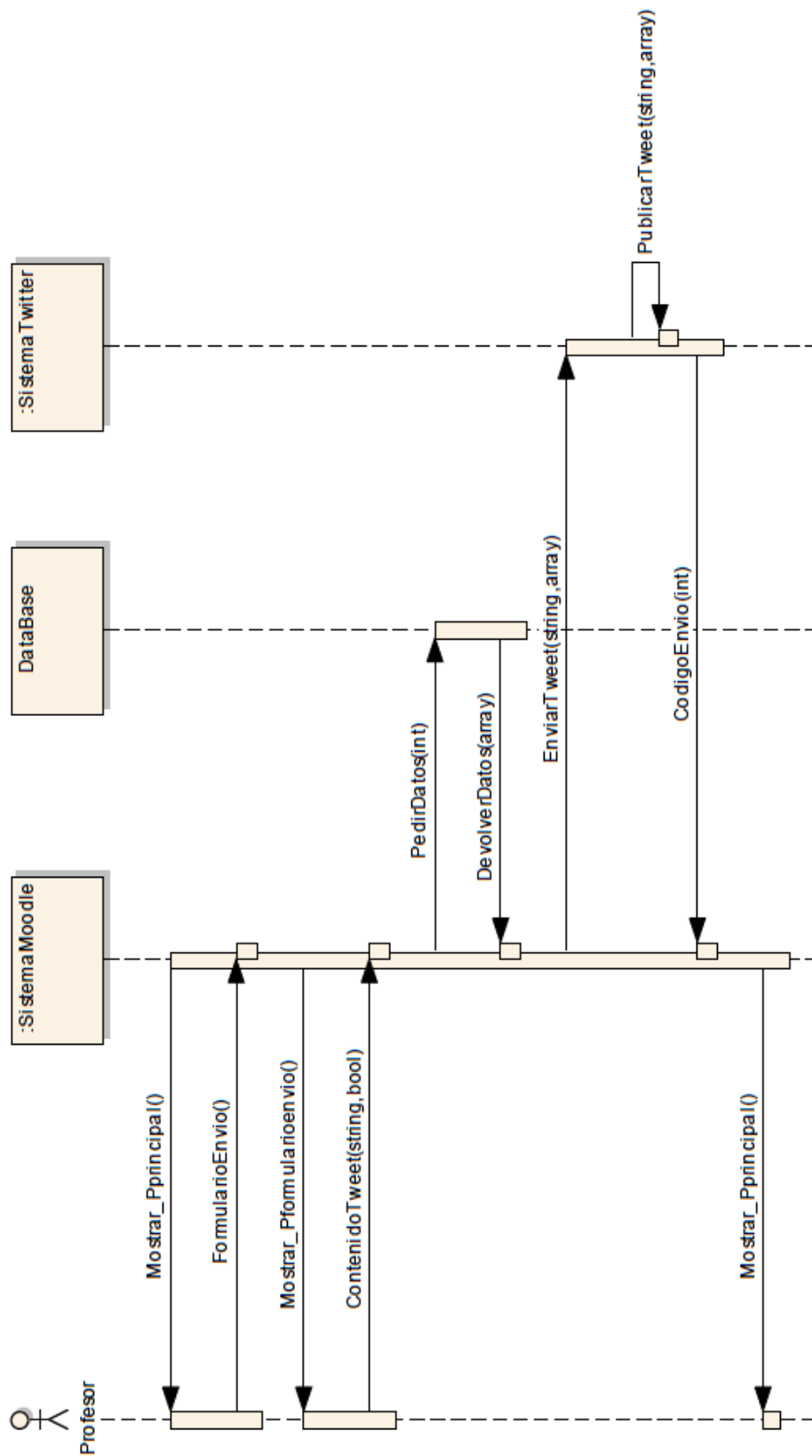


Figura 3.5: Diagrama de secuencia Enviar Tweet

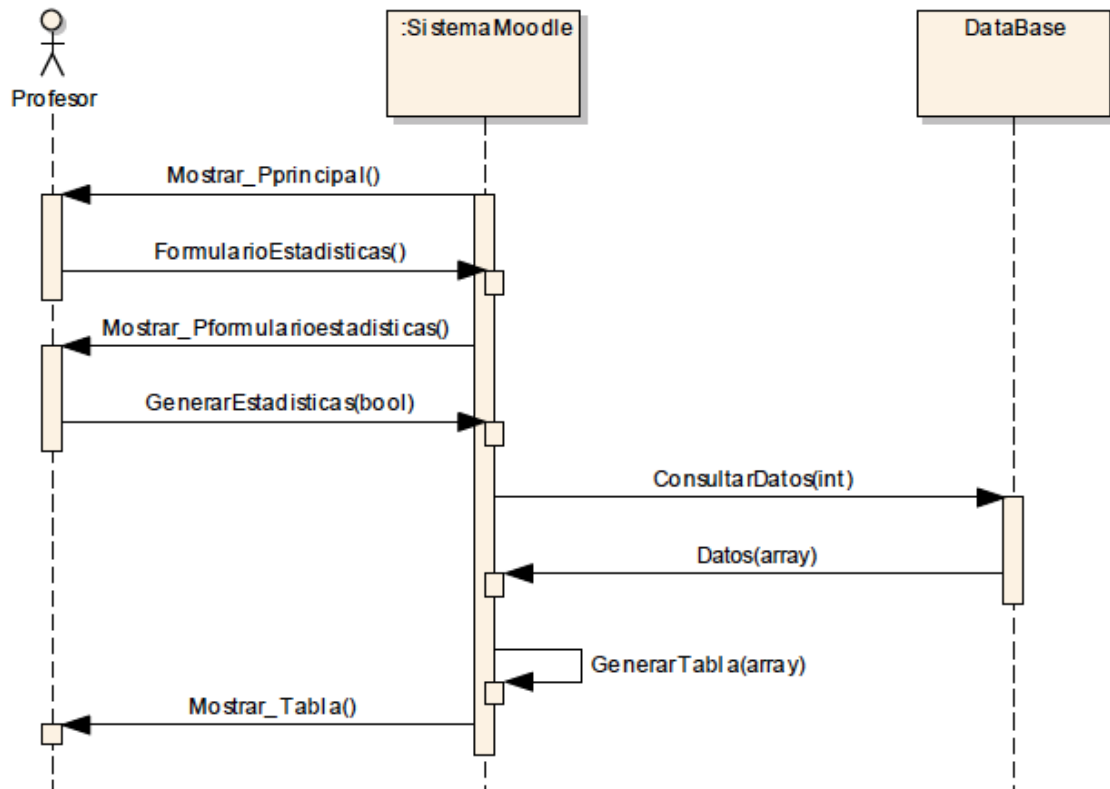


Figura 3.6: Diagrama de secuencia Generar Estadísticas

- `void FormularioEstadisticas()`. El usuario pide al sistema el formulario para generar estadísticas.
- `void Mostrar_Pformularioestadisticas()`. El sistema muestra al usuario el formulario pedido.
- `void GenerarEstadisticas()`. El usuario pide al sistema que genere estadísticas, indicándole al sistema si desea reiniciarlas una vez se generen.
- `void ConsultarDatos(int)`. El sistema consulta a la base de datos los parámetros necesarios para generar las estadísticas del bloque con el id dado.
- `void Datos(array)`. La base de datos devuelve al sistema los parámetros que ha pedido.
- `void GenerarTabla(array)`. El sistema construye la tabla de estadísticas con los datos recibidos.
- `void Mostrar_Tabla()`. El sistema muestra al usuario la tabla con las estadísticas que ha solicitado.

3.4.4. Publicar Actividad Reciente

El diagrama de secuencia correspondiente se puede ver en la Figura 3.7. Las operaciones utilizadas son:

- void EjecutarScript(). El usuario ejecuta el script encargado del mantenimiento del sistema.
- void EnviarTweet(string,array). El sistema envía a Twitter el contenido y los parámetros de todos los tweets que haya que publicar.
- void PublicarTweet(string,array). Twitter publica cada tweet recibido.
- voidCodigoEnvio(int). Por cada tweet publicado, Twitter envía al sistema un código de verificación de la petición.
- void Mostrar_log(). El sistema muestra al usuario un log con las acciones realizadas por el script.

3.4.5. Registrar Nombre Usuario

El diagrama de secuencia correspondiente se puede ver en la Figura 3.8. Las operaciones utilizadas son:

- void Mostrar_Pprincipal(). El sistema muestra la página principal donde se encuentra el bloque al usuario.
- void FormularioRegistro(). El usuario pide al sistema el formulario para registrar su nombre de usuario.
- void Mostrar_Pformularioregistro(). El sistema muestra al usuario el formulario pedido.
- void NombreUsuario(string). El usuario envía al sistema una cadena de caracteres que contiene su nombre de usuario en Twitter.
- void Guardar(string,id). El sistema almacena en la base de datos dicho nombre de usuario recibido junto con el id del usuario que lo ha enviado.

3.4.6. Ver Tweets Recientes

El diagrama de secuencia correspondiente se puede ver en la Figura 3.9. Las operaciones utilizadas son:

- void PedirDatos(int). El sistema consulta a la base de datos los parámetros de la cuenta de Twitter asociada al bloque con un id específico.
- void DevolverDatos(array). La base de datos devuelve los parámetros pedidos al sistema.
- void PedirTweetsRecientes(). El sistema consulta los tweets recientes con los parámetros correspondientes a Twitter.
- void TweetsRecientes(). Twitter responde al sistema enviándole los datos relativos a los tweets recientes.
- void Mostrar_Pprincipal(). El sistema muestra la página principal donde se encuentra el bloque con los tweets recientes al usuario.

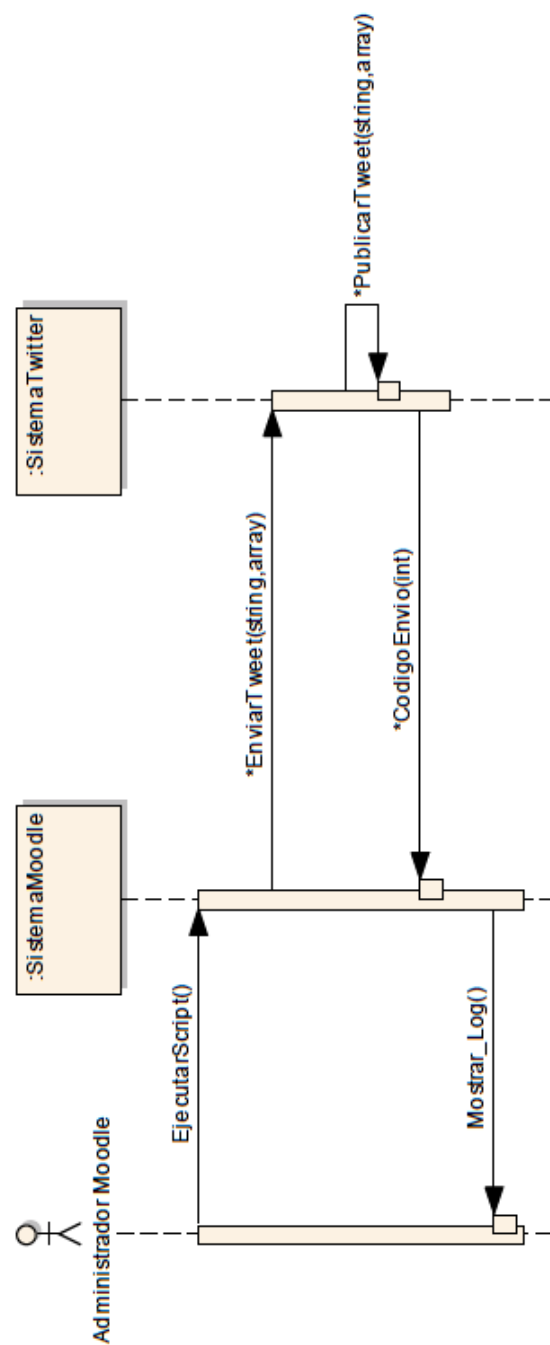


Figura 3.7: Diagrama de secuencia Publicar Actividad Reciente

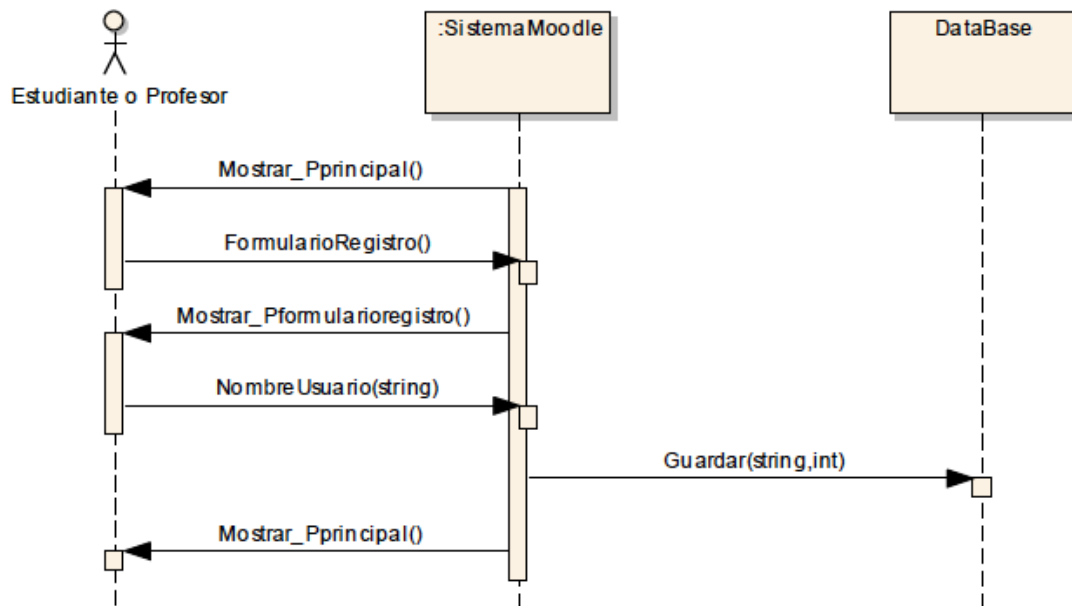


Figura 3.8: Diagrama de secuencia Registrar Nombre Usuario

3.5. Modelo de Interfaz de Usuario

En esta sección se incluye un prototipo de baja fidelidad de la interfaz de usuario del sistema. También se muestra un diagrama de navegación, reflejando la secuencia de pantallas a las que tienen acceso los diferentes roles de usuario y la conexión entre éstas. El modelo de navegación puede verse en la Figura 3.10.

3.5.1. Prototipos de interfaz gráfica

Pantalla Principal

El prototipo de interfaz gráfica para la página principal del bloque puede verse en la Figura 3.11.

Configurar

El prototipo de interfaz gráfica para la página de configuración del bloque puede verse en la Figura 3.12.

Registrar Nombre Usuario

El prototipo de interfaz gráfica para el registro del nombre de usuario puede verse en la Figura 3.13.

Enviar Tweet

El prototipo de interfaz gráfica para el envío de tweets puede verse en la Figura 3.14.

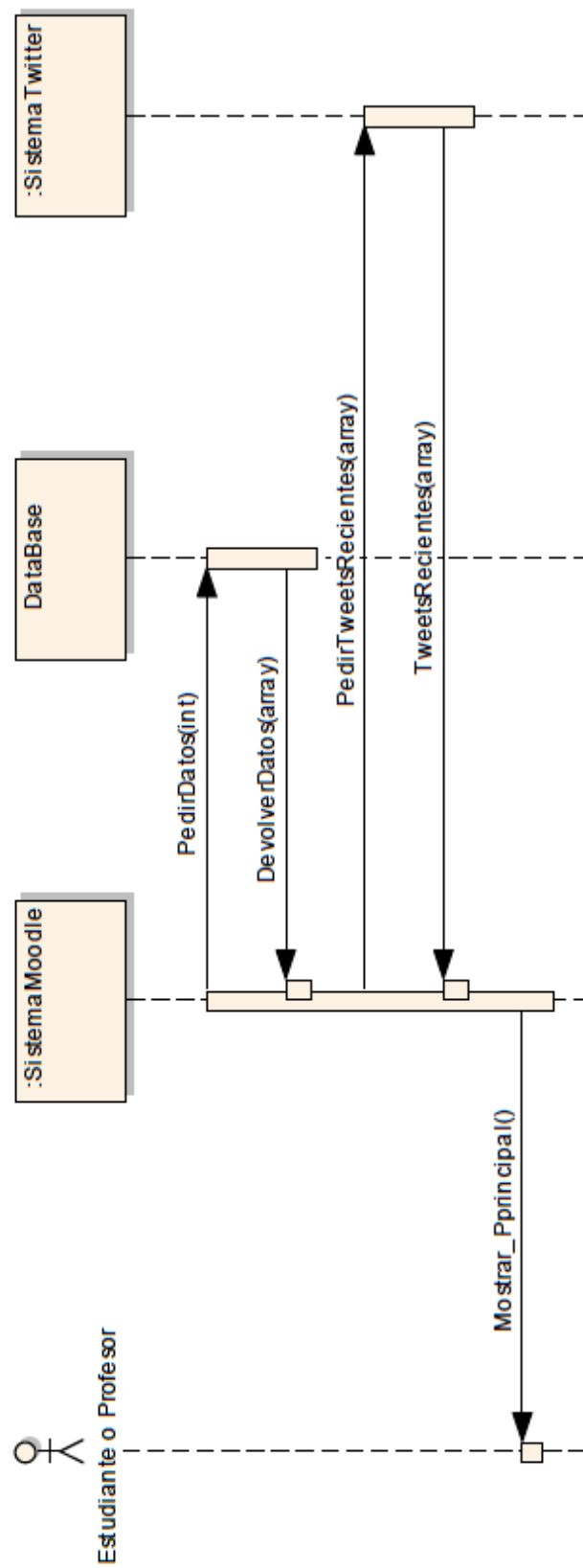


Figura 3.9: Diagrama de secuencia Ver Tweets Recientes

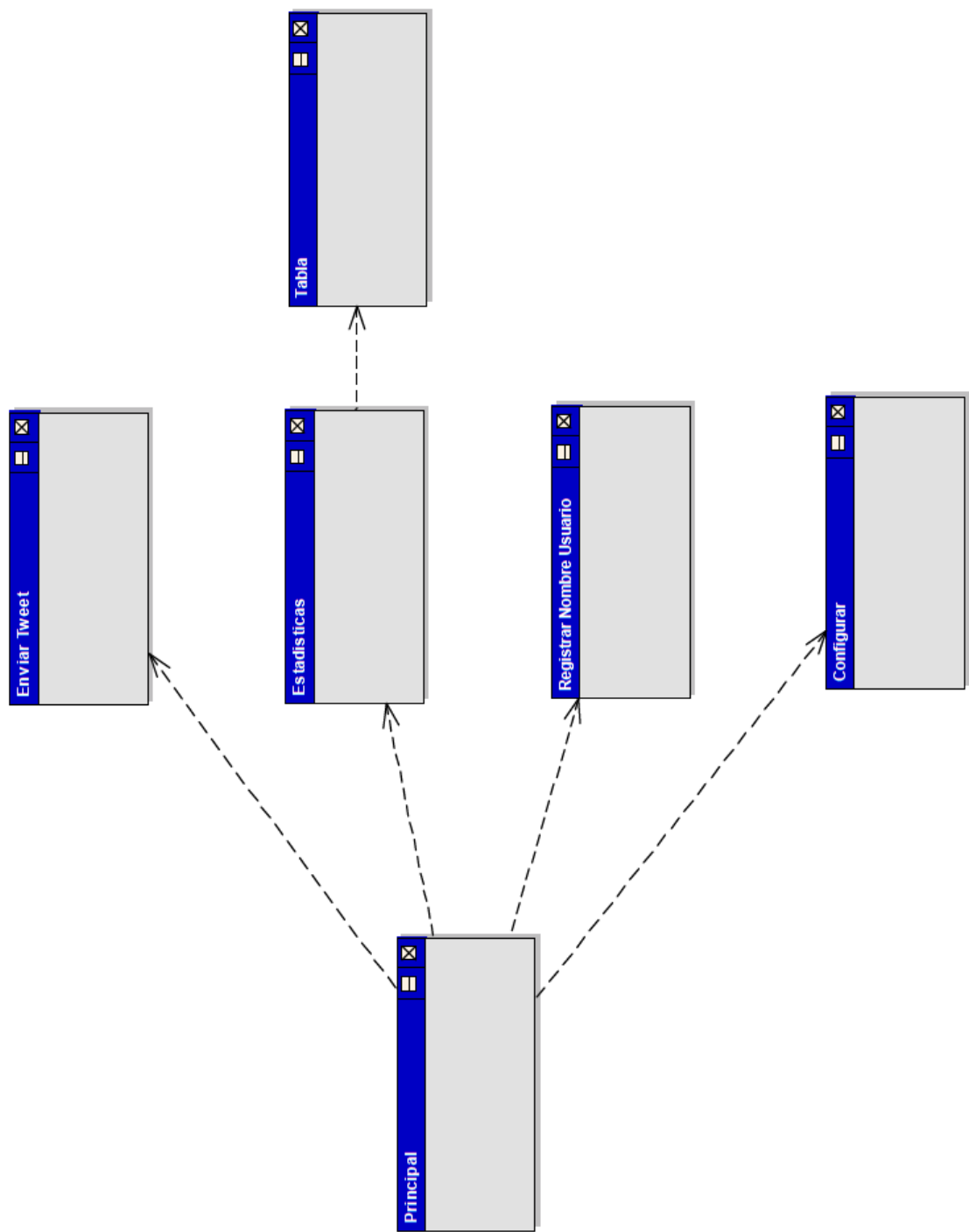


Figura 3.10: Modelo de Navegación de la interfaz gráfica

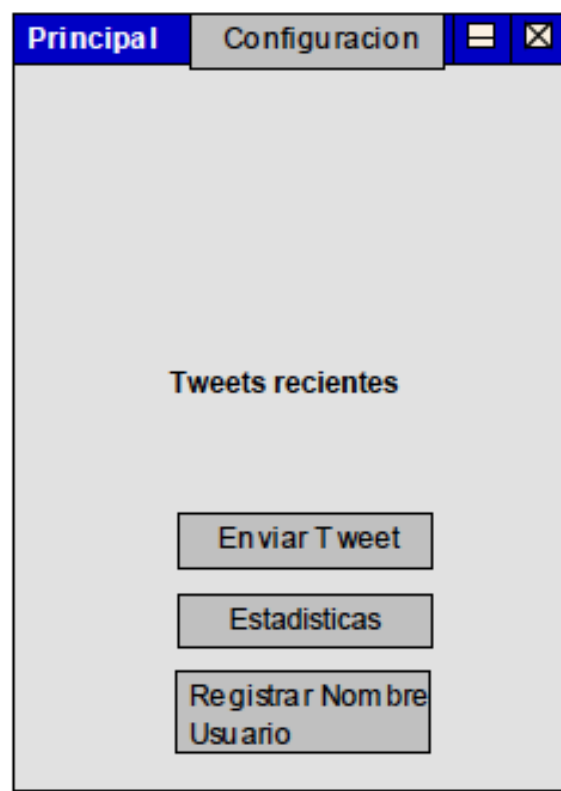


Figura 3.11: Interfaz Gráfica Pantalla Principal

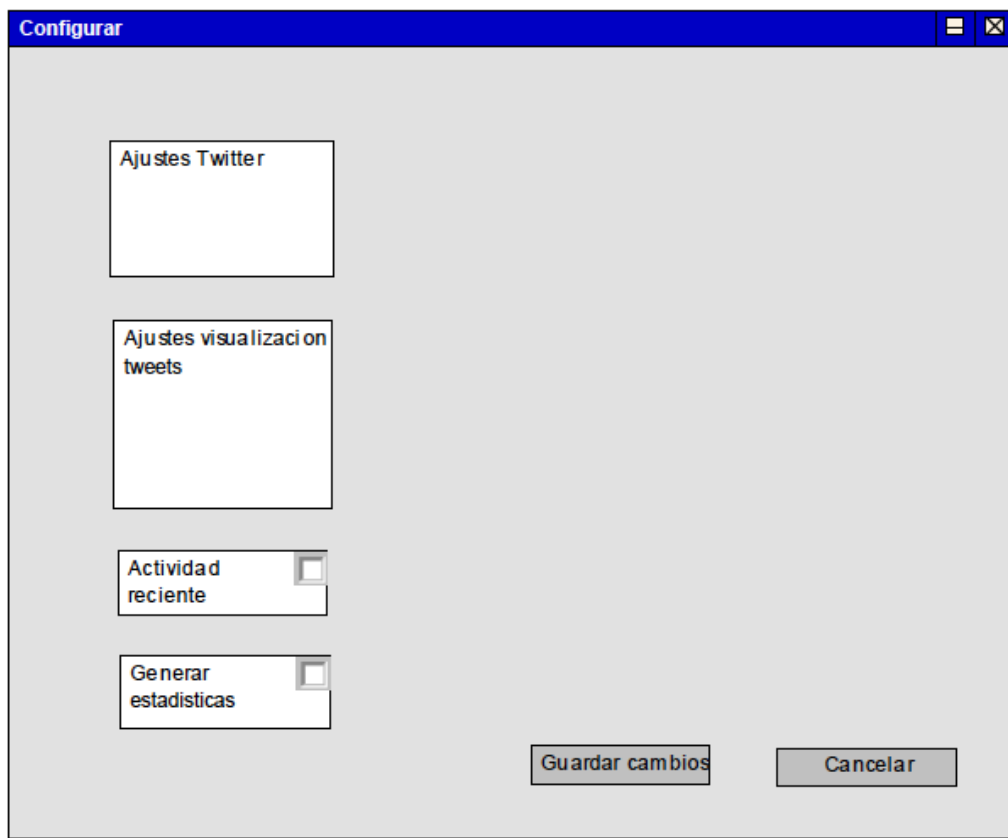


Figura 3.12: Interfaz Gráfica Configurar

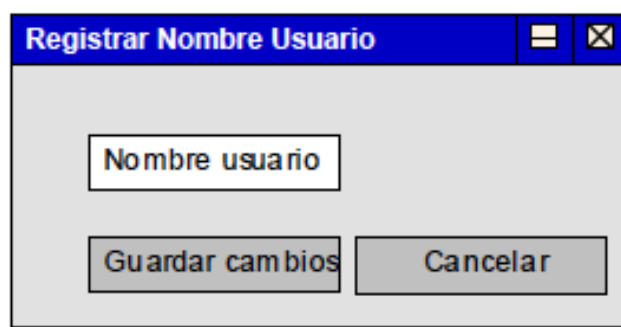


Figura 3.13: Interfaz Gráfica Registrar Nombre Usuario

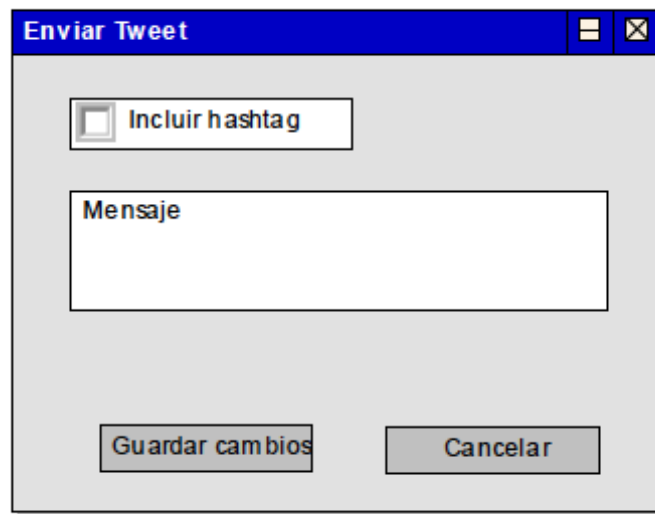


Figura 3.14: Interfaz Gráfica Enviar Tweet

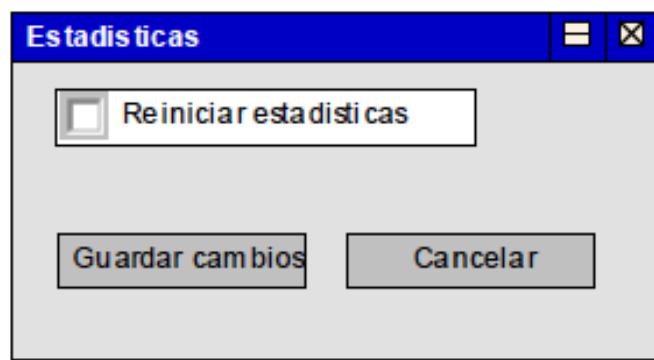


Figura 3.15: Interfaz Gráfica Estadísticas

Generar Estadísticas

El prototipo de interfaz gráfica para el formulario de generación de estadísticas puede verse en la Figura 3.15. Y en la Figura 3.16 se puede visualizar el prototipo de la interfaz gráfica para mostrar las estadísticas.

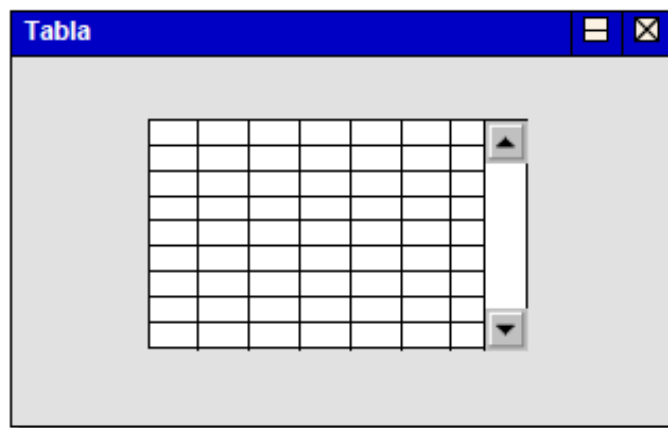


Figura 3.16: Interfaz Gráfica Tabla

Capítulo 4

Diseño del Sistema

En esta sección se recoge la arquitectura general del sistema de información, la parametrización del software base (opcional), el diseño físico de datos, el diseño detallado de componentes software y el diseño detallado de la interfaz de usuario.

4.1. Arquitectura del Sistema

EDUtwitter es un bloque que sirve como extensión al sistema Moodle. Por lo tanto, su desarrollo está orientado a la arquitectura de dicha plataforma. Moodle ha sido desarrollado utilizando LAMP. Es un acrónimo usado para describir un sistema de infraestructura de internet que usa las herramientas:

- Linux (como sistema operativo).
- Apache (como servidor web).
- MySQL (como gestor de base de datos).
- PHP (como lenguaje de programación).

En algunos casos, los sistemas pueden ser intercambiados, usar MariaDB como gestor de base de datos y Perl o Python como lenguajes de programación. Moodle utiliza los primeros descritos. En la Figura 4.1 se puede ver la arquitectura de LAMP usada en un diagrama. Aunque este conjunto de software no fue diseñado específicamente para trabajar entre sí, la combinación se ha hecho muy popular por su bajo coste de adquisición y fácil acceso de sus componentes.

Moodle funciona con una arquitectura bastante usual en las aplicaciones web. Realiza las peticiones a un servidor web, que a su vez utiliza PHP para hacer las peticiones al gestor de base de datos, y a través del servidor web de nuevo muestra la respuesta. Todo esto está representado de forma gráfica en la Figura 4.2.

4.1.1. Arquitectura Física

En este apartado, describimos los principales elementos hardware que forman la arquitectura física de nuestro sistema, recogiendo por un lado los componentes del entorno de producción y los componentes de cliente.

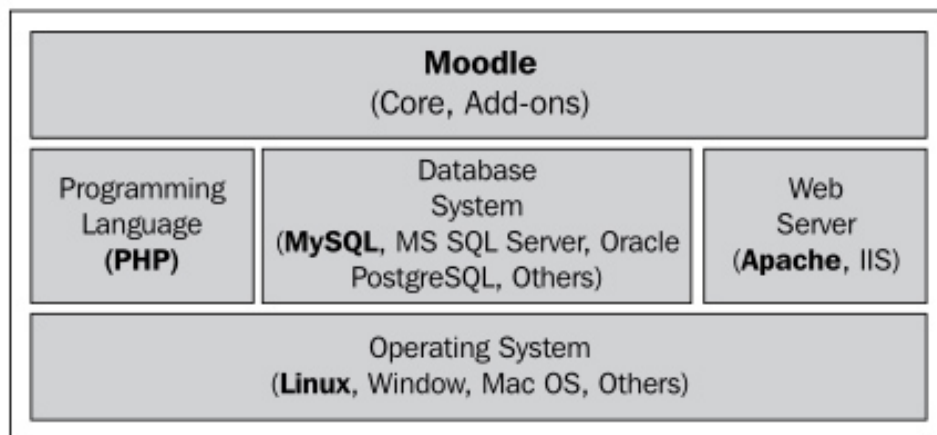


Figura 4.1: Arquitectura LAMP

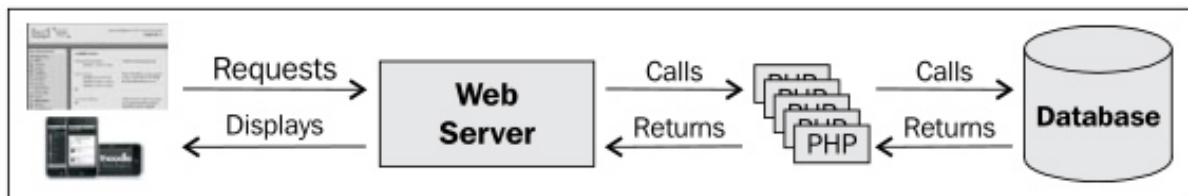


Figura 4.2: Arquitectura Moodle

Entorno de producción

Componentes hardware

Se ha utilizado una sola máquina con las siguientes características:

- Asus EEE PC 1000H.
- Procesador 1.60GHZ dual core.
- 1GB de memoria RAM.

Componentes software

La máquina de producción contaba con el siguiente software:

- Ubuntu 12.04 LTS (Sistema operativo).
- Emacs como editor de texto.
- Plataforma Moodle 2.2.1 para pruebas.
- Apache como servidor web.
- MySQL como gestor de base de datos.
- EDUtwitter, que es la aplicación desarrollada, escrita en PHP.

Entorno de cliente

Componentes hardware

Si se trata del servidor que va a alojar la plataforma Moodle, necesita 1 GB de RAM por cada 10 ó 20 usuarios concurrentes haciendo peticiones al servidor quiera soportar y al menos 5 GB de almacenamiento. Si se trata únicamente del usuario que accede al sistema que está en otro servidor, no requiere ningún requisito de hardware especial.

Componentes software

Si se trata del servidor que va a alojar la plataforma Moodle, necesita todo el software descrito en el Manual de Instalación y Explotación del apéndice. Si se trata únicamente del usuario que accede al sistema que está en otro servidor, sólo necesita conexión a internet y una navegador para acceder a la aplicación web.

4.1.2. Arquitectura Lógica

En la Figura 4.3 podemos ver de forma gráfica la arquitectura lógica del sistema. Aquí se especifica la forma en que los artefactos software interactúan entre sí para lograr el comportamiento deseado en el sistema. En esta sección se muestra la comunicación entre el software base seleccionado, los componentes reutilizados y los componentes desarrollados para cumplir los requisitos de la aplicación. También, se recogen los servicios de sistemas externos con los que interactúa nuestro sistema.

Software base seleccionado

- La plataforma Moodle. EL proyecto trata sobre la creación de un módulo, concretamente un bloque para este sistema. Así que sería el software base del que partimos, porque aunque sea de forma modular, no estamos haciendo otra cosa que ampliarlo.

Componente desarrollado

- El bloque EDUtwitter. Utilizado para ser intermediario entre un LMS como es Moodle y una red social como es Twitter.

Componente reutilizado

- El bloque searchtwitter. Aunque quedó obsoleto con la nueva versión de la Rest API de Twitter, este bloque mostraba los tweets recientes. Se ha reutilizado la función de búsqueda en twitter, adaptándola a la nueva Rest API 1.1.

Servicio utilizado

- Rest API 1.1 de Twitter. Usando la librería Twitter-API-PHP, se ha tenido acceso a esta API para obtener y enviar los datos necesarios y desarrollar las funcionalidades que tiene este proyecto.

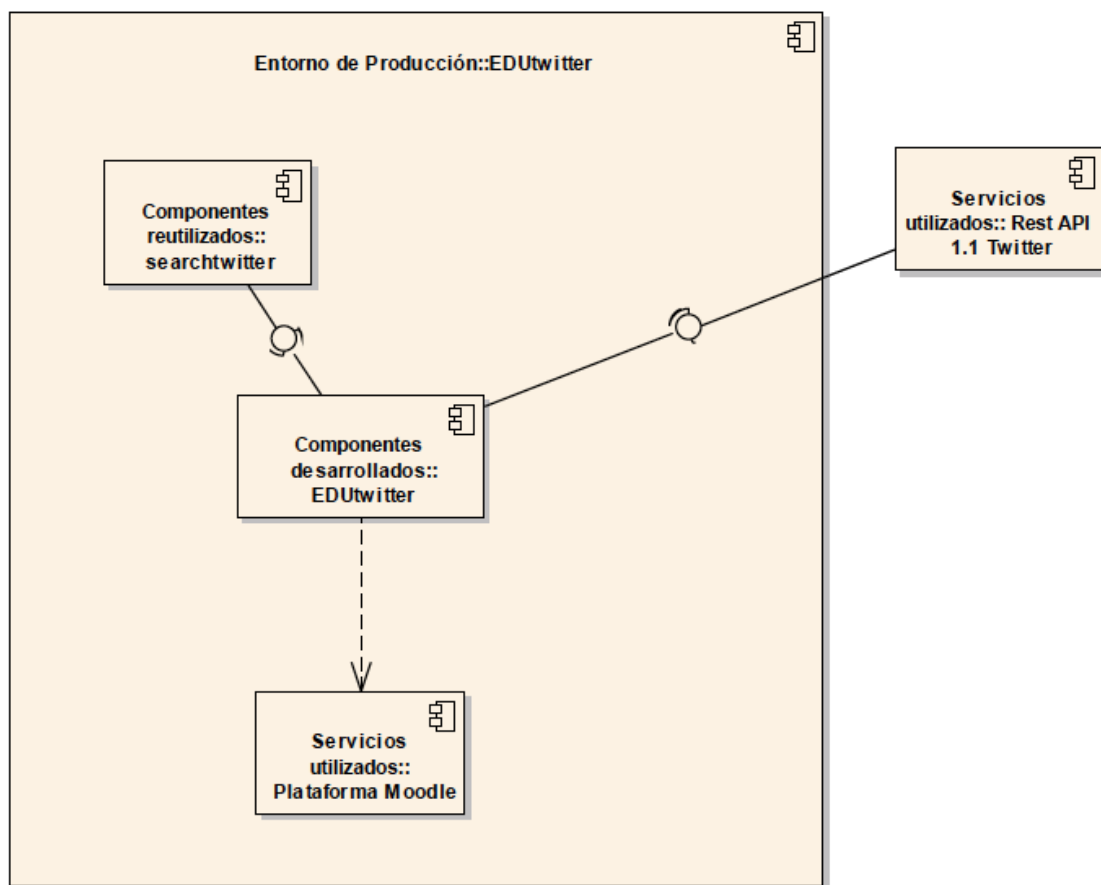


Figura 4.3: Arquitectura Lógica

4.2. Diseño Físico de Datos

En esta sección se define la estructura física de datos que utilizará el sistema, a partir del modelo de conceptual de clases. Este proyecto trabaja junto con la base de datos propia de Moodle, que por su extensión y por no formar parte directamente de este desarrollo no detallamos en este documento. Únicamente hemos simplificado una de las tablas de la base de datos propia de Moodle, la tabla `mdl_course`, ya que las tablas creadas específicamente de este bloque tienen relación de claves foráneas con ésta. Se puede observar el diagrama de diseño en la Figura 4.4. A continuación, detallamos los atributos de las tres tablas creadas, añadidas a la base de datos de Moodle tras la instalación del bloque:

4.2.1. `mdl_block_edutwitter_users`

Esta tabla lleva el registro de los nombres de usuarios en Twitter registrados en el caso de uso “Registrar Nombre Usuario”. Sus atributos son:

- `id` (INTEGER). Identificador del registro, clave primaria.
- `blockid` (BIGINT). Identificador del bloque correspondiente.
- `userid` (BIGINT). Identificador del usuario registrado.
- `username` (VARCHAR). Nombre del usuario registrado.
- `userlast` (VARCHAR). Apellidos del usuario registrado.
- `twitteruser` (VARCHAR). Nombre en Twitter del usuario registrado.
- `courseid` (BIGINT). Identificador del curso correspondiente, clave foránea.
- `list_twitter` (BOOL). Para identificar si ha sido incluido en la lista de alumnos de Twitter o no.

4.2.2. `mdl_block_edutwitter_params`

Esta tabla lleva el registro del último tweet que se revisó, la última vez que se miró la actividad reciente y la última vez que se contaron tweets. Esto es necesario para no publicar la misma actividad reciente y no contar dos veces el mismo tweet a la hora de hacer las estadísticas.

- `id` (INTEGER). Identificador del registro, clave primaria.
- `courseid` (BIGINT). Identificador del curso correspondiente, clave foránea.
- `last_tweet`. Identificador del último tweet revisado.
- `time_recent`. Última ejecución de la publicación de actividad reciente.
- `time_count`. Última ejecución del recuento de tweets.

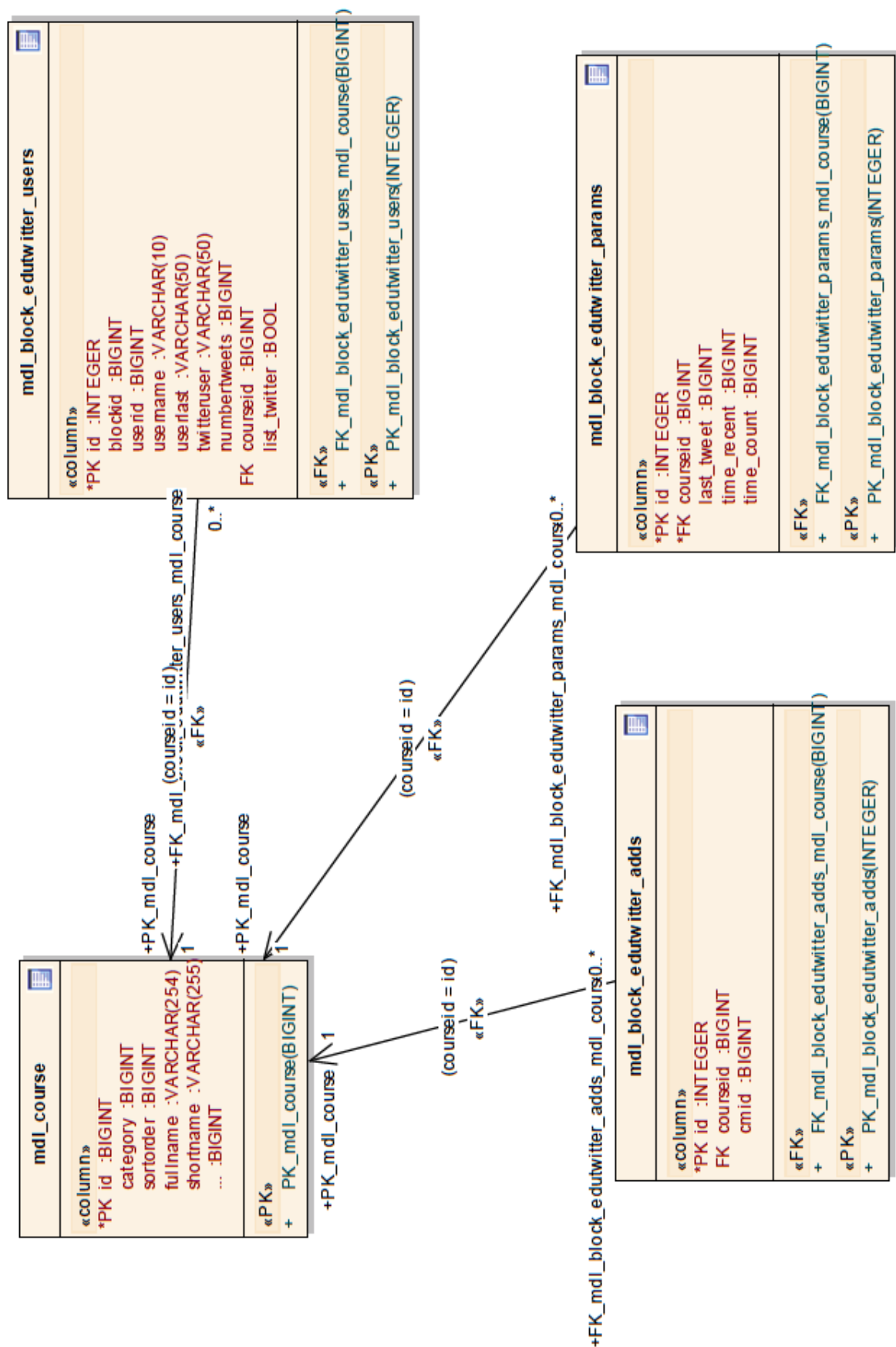


Figura 4.4: Diseño de la Base de Datos

4.2.3. mdl_block_edutwitter_adds

Esta tabla lleva el registro de las actividades y recursos que se han añadido de forma oculta, para que se puedan publicar como actividad reciente una vez se activen en el curso.

- id (INTEGER). Identificador del registro, clave primaria.
- courseid (BIGINT). Identificador del curso correspondiente, clave foránea.
- cmid (BIGINT). Identificador del módulo oculto del curso.

4.3. Análisis de las principales clases de la aplicación

Las principales clases que forman parte de esta aplicación son:

- Clase block_edutwitter. Esta es la clase principal. Hereda de la clase “block_base” del núcleo de Moodle. Es la que se encarga de la apariencia del bloque en la página principal, de enlazar al resto de las funcionalidades y del método que se ejecutará con el script de mantenimiento del sistema completo. Sus métodos son:
 - init(). Pone el título al bloque.
 - get_content(). Gestiona el contenido del bloque. Realiza la llamada a la función que escribe los tweets recientes y controla el rol del usuario para poner unos enlaces u otros, dependiendo si es profesor o alumno.
 - specialization(). Se encarga del contenido del bloque cuando aún no está configurado para mostrar lo anterior.
 - cron(). Es el método que se ejecuta junto con el script de mantenimiento del sistema, que debe ser ejecutado por el administrador del servidor periódicamente.
- Clase block_edutwitter_edit_form. Esta clase hereda de “block_edit_form” de Moodle. Es la que gestiona los parámetros que se configuran en la pantalla de configuración del bloque.
- Clase sendtweet_form. Hereda de la clase “moodleform” de Moodle. Es la clase que gestiona la definición del formulario para enviar tweets.
- Clase statistics_form. Hereda de la clase “moodleform” de Moodle. Es la clase que gestiona la definición del formulario para generar las estadísticas.
- Clase submit_form. Hereda de la clase “moodleform” de Moodle. Es la clase que gestiona la definición del formulario para enviar el nombre de usuario en Twitter.
- Clase TwitterAPIExchange. Es la clase que se encarga de la implementación de la Rest API 1.1 de Twitter para el lenguaje de programación PHP.

4.4. Diseño detallado de la Interfaz de Usuario

En esta sección se detallarán las interfaces entre el sistema y el usuario, incluyendo unas imágenes de alta fidelidad con el diseño de la IU.

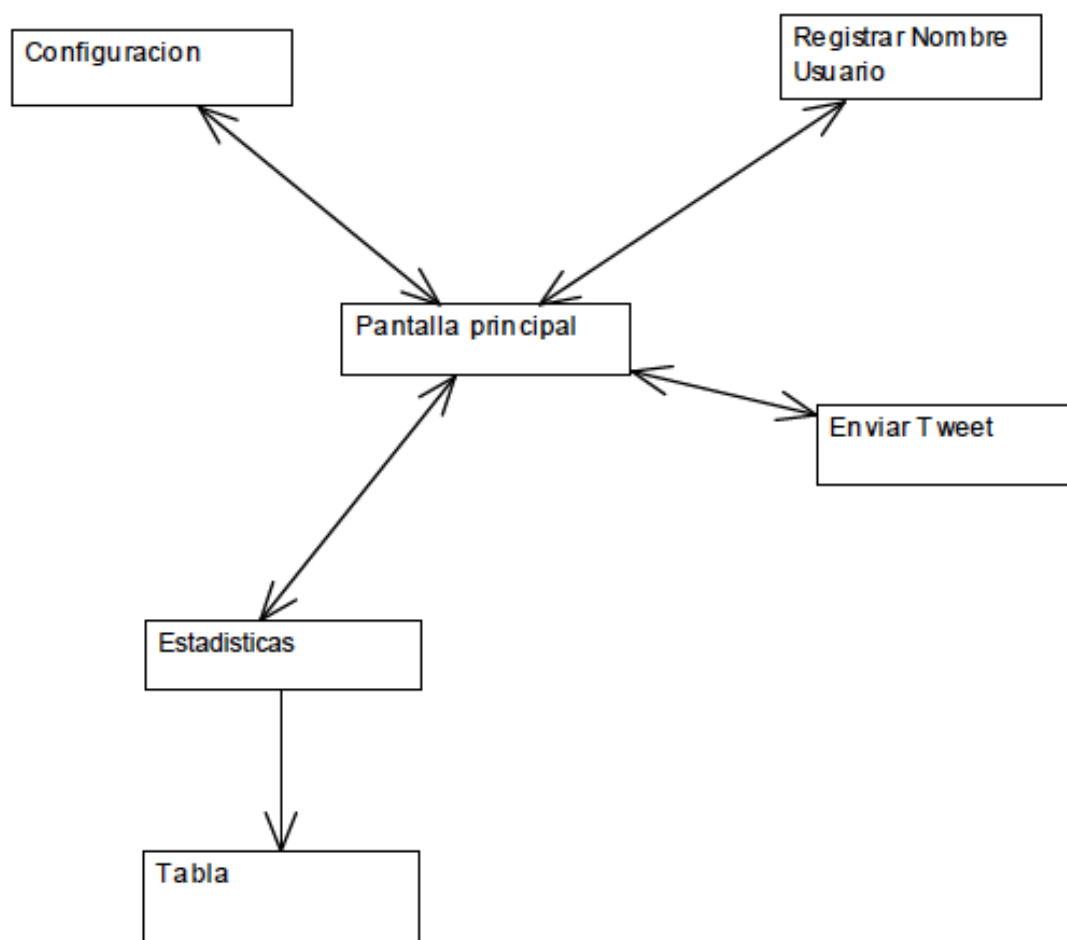


Figura 4.5: Diagrama de navegabilidad

4.4.1. Diagrama de interacción

En la Figura 4.5 se puede ver la navegabilidad entre las diferentes pantallas.

4.4.2. Interfaz

Visión del bloque para alumnos

La podemos ver en la Figura 4.6.

Visión del bloque para profesores

La podemos ver en la Figura 4.7.



Figura 4.6: Bloque con el rol de estudiante



Figura 4.7: Bloque con el rol de profesor

Capítulo 5

Construcción del Sistema

Este capítulo trata sobre todos los aspectos relacionados con la implementación del sistema en código, haciendo uso de un determinado entorno tecnológico.

5.1. Entorno de Construcción

En esta sección se debe indicar el marco tecnológico utilizado para la construcción del sistema: entorno de desarrollo (IDE), lenguaje de programación, herramientas de ayuda a la construcción y despliegue, control de versiones, repositorio de componentes, integración continua, etc.

5.1.1. Nivel de presentación

HTML

HTML son las siglas de HyperText Markup Language, o lenguaje de marcas de hipertexto en castellano. Se trata de un lenguaje de marcado para la elaboración de páginas web. Define una estructura básica con un código para la definición del contenido de éstas. La W3C es la organización encargada del estándar. Este lenguaje de marcado es interpretado luego por los navegadores web, que implementan la forma de mostrar ese contenido con pequeñas diferencias. Dado que la plataforma Moodle es una aplicación web que utiliza este lenguaje para estructurar el contenido, de la misma forma procede el bloque EDUtwitter.

CSS

CSS son las siglas de Cascading Style Sheets, u hojas de estilo en cascada en castellano. Se trata de un lenguaje de hojas de estilo usado para describir la presentación semántica (el aspecto y formato) de un documento escrito en lenguaje de marcas. Complementa al documento HTML dando forma al contenido. El proyecto contiene un archivo escrito en este lenguaje para dar formato a los tweets recientes que se visualizan en el bloque.

5.1.2. Nivel de aplicación

PHP

PHP es un lenguaje de programación del lado del servidor, diseñado para generar contenido web de forma dinámica. Es uno de los pioneros en los que a este tipo de programación se refiere, creado por Rasmus Lerdorf en 1995. El código es interpretado por un servidor web con un

módulo de procesador de PHP que genera la página Web resultante. PHP puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo, forma parte del software libre publicado bajo la licencia PHP. Es el lenguaje de programación usado por la plataforma Moodle, dado su amplia aceptación. Por lo tanto, el proyecto también ha tenido que ser programado en este lenguaje usando un estilo de código particular para integrarse en el sistema.

5.1.3. Nivel de persistencia

MySQL

MySQL es un sistema de gestión de base de datos relacional. MySQL AB desarrolla este sistema como software libre en un esquema de licenciamiento dual, siendo GNU GPL excepto para aquellas empresas que quieran incorporarlo en productos privativos. Es probablemente el más popular en su género, usado por sitios web como Wikipedia, Google, Facebook, Twitter, Flickr y YouTube. Es la opción escogida para la instalación de la plataforma Moodle, que necesita un sistema de gestión de base de datos para funcionar, por su extensa documentación y soporte.

5.1.4. Herramientas

Apache

Apache[4] es un servidor web HTTP de código abierto para plataformas Unix. Se desarrolla dentro del proyecto HTTP Server de la Apache Software Foundation. Tiene una amplia popularidad en la red, siendo el más usado desde 1996. Ofrece características altamente configurables, bases de datos de autenticación y negociado de contenido. Ha sido utilizado y necesario para la instalación de la plataforma Moodle con la que trabajar para el desarrollo del proyecto.

Emacs

Emacs[5] es un editor de texto muy versátil, muy utilizado por programadores y usuarios técnicos. Fue escrito por Richard Stallman junto con Guy Steele en 1975. Hay muchas versiones, la más utilizada es GNU Emacs. Es una herramienta muy útil para la edición de código, ya que se adapta al lenguaje utilizado y aporta mucho soporte a las estructuras de éste a través de indentaciones y coloreado. Ha sido utilizado para escribir todo el código del proyecto, por sus múltiples funciones y las facilidades que aporta a la escritura.

Forja RedIRIS

La forja RedIRIS[6] es uno de los servicios que RedIRIS, la red española para la interconexión de los recursos informáticos de las universidades y centros de investigación, ofrece a toda la comunidad para desarrollar iniciativas libres en el entorno académico-científico universitario. Todo el desarrollo se ha alojado en esta forja, utilizada para poder participar en el Concurso Universitario de Software Libre del curso 2012-2013, en el que este proyecto obtuvo una mención especial en la final nacional.

5.2. Código Fuente

Organización del código fuente, describiendo la utilidad de los diferentes ficheros y su distribución en paquetes o directorios. En la Figura 5.1 puede verse una evolución del número de líneas de

código a lo largo del tiempo por directorio generada con StatSVN. Hay que tener en cuenta en esta figura que la carpeta principal del proyecto “EDUtwitter” fue renombrada al final de su desarrollo por “edutwitter”, ya que para su instalación en el sistema Moodle se requiere que el nombre esté en minúsculas.

- **Carpeta “lang”**

Carpeta que contiene los diccionarios de los distintos idiomas en los que está la interfaz del bloque en Moodle.

- **Carpeta “db”**

Carpeta con la instalación de la base de datos necesaria realizada con el editor XMLDB de Moodle.

- **block_edutwitter.php**

Archivo principal del bloque. Contiene la clase block_edutwitter, las instrucciones de lo que debe mostrarse en el bloque y la actividad que se debe realizar con la ejecución del script cron.php.

- **edit_form.php**

Contiene los elementos que se añaden a los predeterminados en la configuración del bloque.

- **version.php**

Contiene información relativa a la versión necesaria de Moodle para instalar el bloque, la versión de EDUtwitter y la frecuencia con la que se ejecuta dentro del script cron.php.

- **TwitterAPIExchange.php**

Librería^[7] para realizar peticiones a la API 1.1 de Twitter.

- **submit_form.php y view_submit.php**

Contienen los elementos del formulario para enviar el nombre de usuario en Twitter y la forma de procesar esos datos respectivamente.

- **sendtweet_form.php y view_sendtweet.php**

Contienen los elementos del formulario para enviar un tweet y la forma en que se procesa el envío de ese tweet respectivamente.

- **statistics_form.php y view_statistics.php**

Contienen los elementos del formulario para generar estadísticas y la forma en que se crea la tabla y se muestran los datos respectivamente.

- **searchtwitter.php y styles.css**

Contienen la función para buscar y mostrar los tweets recientes con el hashtag del curso y los estilos css en que se muestran respectivamente.

- **split_tweet.php**

Contiene la función que hace posible dividir un contenido de más de 140 caracteres en varios tweets y publicarlos.

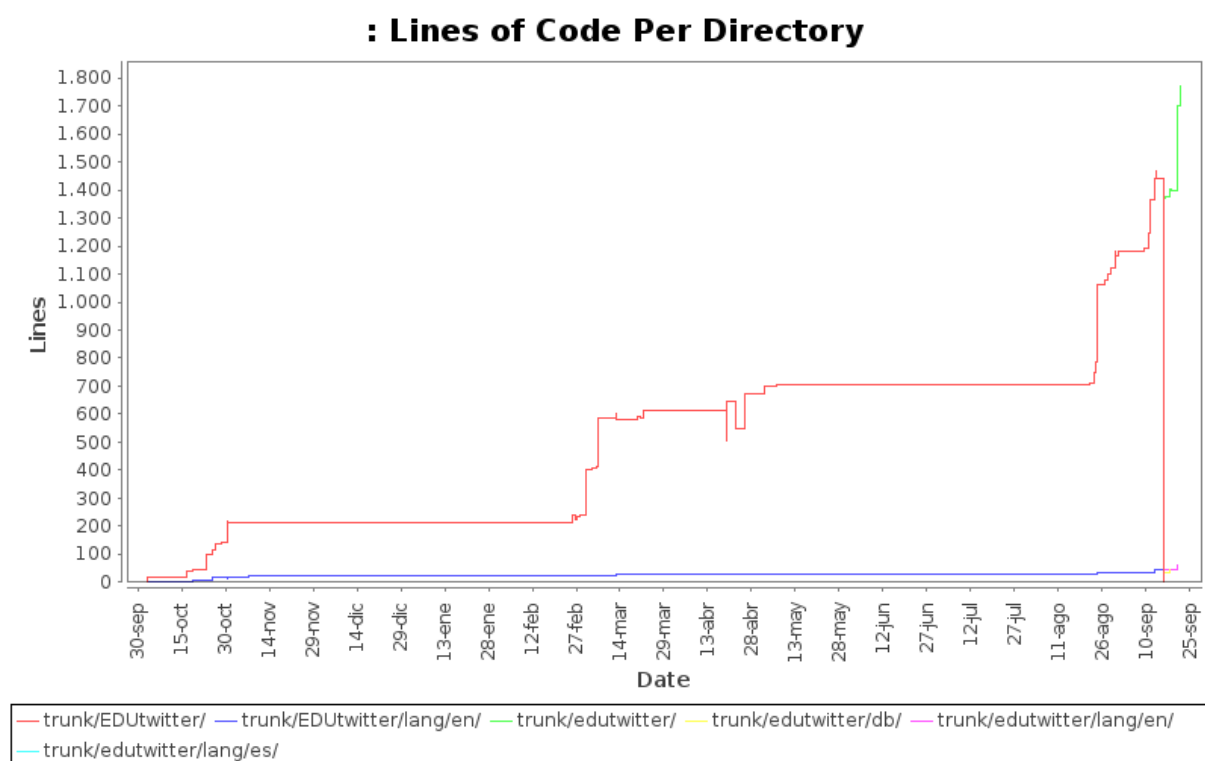


Figura 5.1: Líneas de código a lo largo del tiempo

■ **count_tweets.php**

Contiene la función que cuenta los tweets con el hashtag del curso publicados por los alumnos para la generación de estadísticas.

5.3. Scripts de Base de datos

Organización del código fuente, describiendo la utilidad de los diferentes ficheros y su distribución en paquetes o directorios.

5.3.1. **install.xml**

El bloque EDUtwitter trabaja directamente integrado con la base de datos propia del sistema Moodle. Únicamente requiere de tres tablas más para su funcionamiento. Todas se instalan automáticamente al instalar el bloque en la plataforma a través del archivo “install.xml” que se encuentra en la carpeta “db” del bloque. Las tablas son:

- **block_edutwitter_users.** Esta tabla almacena los nombres de usuarios en Twitter de los usuarios de la plataforma, relacionándolos con el id de usuario y el nombre real.
- **block_edutwitter_params.** Esta tabla almacena los parámetros de la última vez que se publicó la actividad reciente y se contaron tweets para cada instancia, con el objetivo de que no haya duplicaciones.

- **block_edutwitter_adds.** Esta tabla es necesaria para que no se publique en la actividad reciente que se ha añadido un recurso si éste se ha añadido de forma oculta.

Capítulo 6

Pruebas del Sistema

En este capítulo se presenta el plan de pruebas del sistema de información, incluyendo los diferentes tipos de pruebas que se han llevado a cabo, ya sean manuales (mediante listas de comprobación) o automatizadas mediante algún software específico de pruebas.

6.1. Estrategia

Todas las pruebas que se han realizado han sido de tipo manual, ya que automatizar unas pruebas era un proceso más complicado que el propio desarrollo por tener que pasar a través del sistema Moodle. Además dada la cantidad de funcionalidades del proyecto, no ha sido necesaria la automatización. Se han realizado pruebas unitarias de aceptación, donde los mismos clientes han evaluado los resultados. Para las pruebas de regresión fue necesario detectar las funciones dependientes de otras. De forma que cuando se modificase alguna que tuviese algo que ver con otra, se volvían a realizar las pruebas de la segunda.

6.2. Entorno de Pruebas

Los requisitos de los entornos hardware y software para realizar las pruebas del sistema son:

6.2.1. Requisitos hardware

- Espacio en disco: 160MB libres (mínimo).
- Memoria RAM: 256MB (mínimo), 1GB o más sería muy recomendado.

6.2.2. Requisitos software

- Un sistema operativo Linux que soporte el siguiente software.
- Un servidor web. Preferiblemente Apache o ISS.
- PHP. La versión mínima permitida es 5.3.2.
- Una base de datos MySQL (versión mínima 5.0.25).
- Plataforma Moodle (versión mínima 2.2.1) instalada y correctamente configurada.

6.3. Roles

Los perfiles participantes de las pruebas han sido de tres tipos:

- Desarrollador del software. El mismo desarrollador ha realizado las pruebas en su propio producto para comprobar que funciona correctamente.
- Supervisores. Los supervisores ha comprobado el funcionamiento del producto a través de demostraciones.
- Testeadores independientes. Expertos en el sistema han realizado las pruebas necesarias al producto para poder ponerlo a disponibilidad pública.

6.4. Niveles de Pruebas

En esta sección se documentan los diferentes tipos de pruebas que se han llevado a cabo.

6.4.1. Pruebas Unitarias

El proyecto se ha ido desarrollando por casos de uso según el análisis. Para cada función que se completaba, se recorrían los distintos escenarios y se comprobaba que tenía el comportamiento adecuado. Esto requería la actualización de la versión del bloque para que el sistema Moodle tuviera en cuenta los cambios realizados.

6.4.2. Pruebas de Integración

En estas pruebas se tenía en cuenta la completa integración con la red social Twitter. Para cada comportamiento en el sistema Moodle, se comprobaba que la cuenta de Twitter asociada al curso se comportaba también de la forma adecuada y no producía fallos de publicación.

6.4.3. Pruebas de Sistema

En esta actividad se realizan las pruebas de sistema de modo que se asegure que el sistema cumple con todos los requisitos establecidos.

Pruebas Funcionales

En estas pruebas se realizó un seguimiento global del funcionamiento. Se instaló un bloque EDUtwitter y se configuró a partir de cero. Posteriormente se crearon usuarios matriculados en el curso que hicieron uso de sus funcionalidades. Luego se realizaron pruebas con cuentas de Twitter que utilizaban el hashtag del curso para comprobar cómo recogía el bloque esos datos. Se comprobó en la base de datos y en la publicación de Twitter que todos los casos de uso y sus escenarios alternativos correspondientes funcionaban correctamente.

6.4.4. Pruebas de Aceptación

Se envió el proyecto para su evaluación por un grupo de expertos para publicarlo en la página web oficial del sistema Moodle y se le mostró su funcionamiento a los supervisores para comprobar que el producto estaba listo para la fase de producción.

6.5. Resolución de errores

En esta sección describimos algunos de los errores más destacados encontrados en las pruebas y las respectivas soluciones implementadas.

6.5.1. Exceder el número de caracteres en un tweet

Descripción

Enviar un tweet a través de la Rest API de Twitter con más de 140 caracteres provoca un error que impide su publicación. Esto proporcionaba errores desde dos funciones distintas. Por un lado, el formulario de “Enviar tweet”. En este caso, el contenido del texto que se enviaba estaba limitado por el propio formulario. Éste no permitía que se procesasen mensajes que sobrepasasen este límite de caracteres. El problema comenzaba cuando se introducía una página web en el mensaje. Twitter codifica por defecto todas las páginas web de un tweet en una url de 22 caracteres. Si se escribía una url inferior a ese número, el formulario lo aceptaba, pero Twitter no lo publicaba. El segundo problema venía por la publicación de la actividad reciente. En ésta, la notificación de un nuevo mensaje en el foro va acompañada en el tweet por el asunto de dicho mensaje. Si se trataba de un asunto considerablemente largo, el conjunto excedía los 140 caracteres y no se publicaba la notificación en Twitter.

Solución implementada

Se creó una función que controlara todos estos aspectos y asegurara la publicación del mensaje. Dicha función intenta publicar el tweet, si recibe un error de sobrepasar el límite de caracteres, divide el tweet en dos. Mantiene en ambos el hashtag del grupo, al primero le añade en el punto de corte unos puntos suspensivos y al segundo les añade esos mismos puntos al inicio. Luego, publica el primero de ellos (comprobado que tiene menos de 140 caracteres) y vuelve a llamarse de forma recursiva con el segundo, por si hubiese que volver a dividirlo.

6.5.2. Publicación de recursos añadidos de forma oculta

Descripción

La publicación de la actividad reciente funciona de forma parecida a la del bloque de actividad reciente. El problema es que dicho bloque muestra la información personalizada al rol del usuario que la está consultando. Es decir, que si un profesor añade un recurso de forma oculta, a éste le aparecería en el bloque de actividad reciente, pero a un estudiante no. La función que publica la actividad reciente en Twitter se ejecuta en un script con privilegios de administrador, por lo que publicaba los recursos que se añadían de forma oculta. Esto era un problema, ya que nos interesaba que sólo se publicase la información que se puede observar con el rol de estudiante.

Solución implementada

La solución fue crear una tabla en la base de datos que controlara este problema. Dado que no se activaba ninguna función especial a vigilar cuando un profesor cambiaba un recurso de oculto a visible, hubo que crearla. En esta tabla se almacena cada recurso o actividad que un profesor ha añadido de forma oculta. Cada vez que se realiza la función de publicar la actividad reciente en Twitter, se recorre esta tabla comprobando que todos los elementos siguen ocultos. Si hay

alguno que en ese momento ya es visible, se publica la notificación en Twitter y se elimina de la tabla el registro.

6.5.3. Búsqueda exacta en Twitter

Descripción

La Rest API de Twitter no incluye ninguna función que cuente tweets (la que existe sólo puede ser usada por desarrolladores de la red social), ni tampoco una búsqueda exacta de un hashtag determinado. Por un lado, la búsqueda de Twitter solo busca coincidencias en la última semana. Además, Twitter no asegura que esa búsqueda proporcione todos los resultados de esa última semana, porque tiene un algoritmo para mostrar solo los más destacados. Esto consistía un grave problema a la hora de contar los tweets de cada estudiante para las estadísticas, que tenía que funcionar de una forma exacta.

Solución implementada

La solución para que la búsqueda mostrase todos los resultados era única: insertar a los alumnos en una lista. Las búsquedas en una lista no tienen limitación temporal y muestra todos los resultados. Así que cada vez que se ejecuta el script de mantenimiento, se añaden todos los usuarios que hayan registrado su nombre de usuario en Twitter desde la última vez que se ejecutó dicho script a una lista Alumnos de la cuenta de Twitter asociada al curso. Luego, se miran todos los tweets de esa lista desde el último que se miró en la última ejecución y para cada uno se identifica el autor y si ha publicado con el hashtag del curso. Si cumple esta última condición, se añade uno más a su contador de tweets.

Parte III

Epílogo

Capítulo 7

Conclusiones

En este último capítulo se detallan las lecciones aprendidas tras el desarrollo del presente proyecto y se identifican las posibles oportunidades de mejora sobre el software desarrollado.

7.1. Objetivos alcanzados

Desde que se planteó este proyecto con unos objetivos definidos, se han ido redefiniendo adaptándolo a las capacidades del sistema. Esta es la descripción de los objetivos que se han conseguido alcanzar:

- Relacionar una cuenta de Twitter con un curso de Moodle. EDUtwitter enlaza un curso de Moodle con una cuenta de Twitter, dando permisos para poder interactuar con la cuenta sin necesidad de acceder a través de la red social. Además, crea y conserva un hashtag sobre el curso para que queden bien identificados los mensajes que se refieran a éste. Se podrá comprobar si existe un uso previo de dicho hashtag antes de comenzar a usarlo, para esto hay que tener en cuenta que la búsqueda de Twitter se limita a los mensajes enviados en la última semana.
- Enviar información a Twitter desde Moodle. EDUtwitter permite que se puedan mandar mensajes como si desde la misma cuenta de Twitter se tratase. Además también se encarga de enviar información sobre la actividad reciente: nuevos alumnos añadidos al curso, nuevos recursos disponibles, novedades en el foro y el envío de tareas. También incorpora un método para que cualquier desarrollador pueda hacer uso del bloque para enviar a Twitter la actividad reciente de su módulo.
- Obtener datos del hashtag del curso. EDUtwitter puede realizar también el proceso inverso, creando una comunicación bidireccional. Es decir, puede obtener información de la cuenta de Twitter. Muestra en el curso de Moodle los últimos mensajes enviados con el hashtag del curso de forma totalmente configurable. Además, almacena información sobre el envío de mensajes relacionados con el curso enviado por los alumnos.
- Generar estadísticas de la participación de los alumnos. EDUtwitter es capaz de recopilar los datos almacenados sobre el envío de mensajes en la red social Twitter relacionados con el curso por parte de los alumnos y elaborar estadísticas que pueden ayudar a los docentes a la hora de evaluarlos.

7.2. Lecciones aprendidas

Este ha sido el primer proyecto serio de software en el que me he visto envuelto. Por lo que me he visto obligado a desarrollar muchas competencias transversales aparte del aprendizaje que tenía que llevar a cabo para poder desarrollar el proyecto en sí.

Las buenas prácticas procedimentales adquiridas han sido:

- Control de versiones. Es la primera vez que me he obligado a llevar un control de versiones para un proyecto en solitario. En este caso he utilizado la forja RedIRIS, que ha contribuido a la mejor planificación de los elementos que quedaban por implementar, así como poder controlar cualquier versión del bloque y su desarrollo. El único problema ha sido las veces que me he visto obligado a trabajar offline, que ha hecho que todos los cambios se hayan hecho de una sola vez en la forja, en vez de paso a paso.
- Documentación. En todos mis proyectos anteriores, la documentación ha sido una asignatura pendiente. La tengo considerada como algo fundamental dentro de un proyecto serio, pero requiere una carga considerable de trabajo. He intentado ir documentando todo lo aprendido en forma de tutorial en un blog. Así también le iba dando visibilidad y difusión al bloque, y podía ayudar en el aprendizaje a otras personas interesadas en el mismo tipo de desarrollo.
- Trabajar de forma modular. Es la primera vez también que participo con mi desarrollo en un proyecto más grande. El bloque EDUtwitter se integra de forma modular con el sistema Moodle. Esto proporciona algunas ventajas, como que puedes utilizar ciertas funciones sin saber cómo están implementadas y aporta una organización parecida en todos los módulos. Aunque también he tenido problemas por querer implementar algo que no se podía hacer de forma modular y necesitaba la modificación del sistema, por lo que he tenido que deshechar esos casos.
- Adecuación del estilo de código. Los desarrollos realizados para el sistema Moodle deben de escribirse con un estilo de código estipulado por la plataforma. Esto garantiza que sea mucho más fácil entender las implementaciones que han realizado otros colaboradores. Ha sido difícil adaptarse en un primer momento a un estilo de escritura de código concreto, pero es un periodo de adaptación que merece la pena a la larga.
- Compartir el desarrollo. He intentado que el bloque tuviera visibilidad desde el primer momento. Para ello lo he presentado al Concurso Universitario de Software Libre (CUSL). Es un concurso de desarrollo de software, hardware y documentación técnica libre en el que pueden participar estudiantes universitarios a nivel nacional. EDUtwitter obtuvo una mención especial en la final que tuvo lugar en Granada. Además de esto, también lo he enviado para su publicación en la página oficial de la plataforma Moodle, donde están alojados el resto de los bloques aprobados. Me pareció muy importante dar a conocer el proyecto para poder escuchar opiniones de la comunidad sobre el desarrollo.

Las tecnologías utilizadas y aprendidas las podemos enumerar:

- PHP. Aunque durante mis estudios he utilizado este lenguaje de programación de una forma muy básica, he necesitado aprenderlo a fondo para poder utilizarlo en el desarrollo, ya que es el lenguaje utilizado por Moodle y sus módulos. Agradezco la inversión de tiempo ya que hoy en día conocer este lenguaje se ha vuelto fundamental en el mundo de la programación.

- Rest API. Es la primera vez que trabajo con el concepto de API de esta forma. En este caso, a través de la API 1.1 de Twitter en su versión Rest, que ha hecho que comprenda perfectamente el concepto y su funcionamiento. Además, he ido probando varias versiones de las librerías PHP que hacen uso de la API hasta dar con la más correcta para mi proyecto.
- Plataforma Moodle. Durante el desarrollo del proyecto, he podido aprender cómo instalar, administrar y usar la plataforma Moodle. También evidentemente cómo contribuir con la comunidad y de manera específica la forma de crear un bloque para su uso por parte de los usuarios. Considero que esto me será muy útil dada la trayectoria orientada hacia la docencia que pretendo seguir.
- Latex. Para la realización de la memoria he utilizado Latex por la facilidad que aporta a la organización del contenido. He tenido que aprender a utilizarlo y me parece una herramienta muy productiva y de fácil uso una vez que te adaptas para realizar cualquier tipo de documento de texto que vaya a ser publicado.

7.3. Trabajo futuro

En esta sección, se presentan las diversas áreas u oportunidades de mejora detectadas durante el desarrollo del proyecto y que podrán ser abarcadas en futuras versiones del software.

- Múltiples hashtags específicos: En un principio se pensó en dar la posibilidad de crear varios hashtags para las distintas tareas y para los distintos recursos. De forma que además del hashtag propio del grupo, se añadiera en el mensaje el hashtag de lo que se está hablando específicamente dentro de éste. Se desechó por no poder generar contenido en el formulario de configuración de forma dinámica dependiendo de las tareas activas. Pero habría que buscar la forma de implementarlo.
- Nuevas estadísticas: Añadir nuevas opciones para generar otras estadísticas ayudará a los docentes a recabar información para la evaluación. Pueden ir orientadas a los múltiples hashtags específicos para conocer los recursos que más éxito tienen entre los alumnos.
- Personalización de la actividad reciente: Ampliar la configuración para señalar exactamente que es lo que queremos que se publique como actividad reciente y que no, ayudará a no colapsar de mensajes que carezcan de la utilidad que buscamos en la cuenta del curso.
- Evaluación a partir de rúbrica: Creación de una rúbrica que permita ponderar la nota relativa a los mensajes en Twitter de los alumnos puede ser una buena herramienta para fortalecer los criterios de evaluación.
- Evaluación de calidad por compañeros: Para evitar el envío de mensajes sobre el curso sin ninguna aportación de calidad, se puede implementar un sistema para que los alumnos evalúen las aportaciones de sus compañeros en función de lo útiles que les han sido.

Apéndice A

Manual de implantación y explotación

Las instrucciones de instalación y explotación del sistema se detallan a continuación.

A.1. Introducción

EDUtwitter es un bloque para la plataforma educativa Moodle, creado para poder aprovechar las características que nos ofrece la red social Twitter para mejorar la comunicación y complementar la evaluación entre los participantes de un curso.

La comunicación entre profesores y entre alumnos y profesores en el aula virtual puede resultar lenta en algunos casos en los que la llegada de información debe ser rápida y eficaz, como el retraso o cambio de aula de una clase. Además, el desarrollo de las comunicaciones móviles ha propiciado el intercambio de mensajes cortos y la interfaz de Moodle no está optimizada para estos dispositivos.

Actualmente el uso de Twitter está muy extendido y se adapta a la inmediatez de las aplicaciones móviles en el bolsillo. EDUtwitter aprovecha esto y lo integra junto con un curso del aula virtual para que las comunicaciones, mensajes, opiniones y demás lleguen al conocimiento de los profesores y estudiantes lo antes posible. Además, recopila el uso que hacen los alumnos de esta red social para hablar del curso y así complementar su evaluación final.

A.2. Requisitos previos

Requisitos hardware y software para la correcta instalación del sistema. Son los mismos que los necesarios para instalar el entorno Moodle.

A.2.1. Requisitos hardware

- Espacio en disco: 160MB libres (mínimo) más tanto como se necesite para almacenar los recursos y materiales. 5GB es probablemente un mínimo realista.
- Espacio para copias de seguridad: Al menos el mismo espacio que el anterior (remoto preferiblemente) para almacenar una copia de seguridad del sitio.
- Memoria RAM: 256MB (mínimo), 1GB o más sería muy recomendado. La regla general es que Moodle puede soportar de 10 a 20 usuarios concurrentes por cada 1GB de RAM, pero variará dependiendo del hardware específico, de la combinación de software y del tipo de

uso. Usuarios concurrentes quiere decir realizando al mismo tiempo peticiones al servidor, no se refiere al número de usuarios que tienen la sesión iniciada.

A.2.2. Requisitos software

- Un sistema operativo. Cualquiera que soporte el siguiente software. Linux y Windows suelen ser las opciones comunes y las que tienen el mejor soporte.
- Un servidor web. Preferiblemente Apache o ISS.
- PHP. La versión mínima permitida es 5.3.2.
- Una base de datos. MySQL (versión mínima 5.0.25) y PostgreSQL (versión mínima 8.3) son las bases de datos de desarrollo primarias, las más probadas y tienen una documentación y soporte extensos. Oracle (versión mínima 10.2) y MSSQL (versión mínima 9.0) están totalmente soportados, pero la documentación no es tan buena. Se necesitará la extensión PHP apropiada para la base de datos elegida.

A.3. Inventario de componentes

Lista de los componentes hardware y software que se incluyen en la versión del producto.

A.3.1. Twitter-API-PHP

Se trata de una librería simple para acceder a la API versión 1.1 de Twitter a través de PHP 5.3.10 utilizando OAuth como método de autenticación (obligatorio a partir de la versión 1.1 de la API). Es la librería utilizada para realizar todas las peticiones a la API de Twitter que se hacen a través del bloque. Tiene licencia GPL.

A.3.2. Twitter Search Block

Para buscar y mostrar los tweets recientes con el hashtag del curso, se ha tomado como referencia la función searchtwitter del bloque para Moodle “Twitter Search Block”. Aunque se ha modificado casi completamente por tener algunos errores y por tenerse que adaptar a la versión 1.1 de la API de Twitter, ya que estaba pensado para la versión 1.0 que no requería autenticación para este tipo de petición. Tiene licencia GPL.

A.4. Procedimientos de instalación

Para la instalación de EDUtwitter, es necesario antes tener instalada una versión de Moodle (mínimo la versión 2.2.1).

Primero, describimos la instalación del sistema Moodle en una distribución Linux:

1. Instalar MySQL.

Escribimos en la consola lo siguiente para descargarlo:

```
sudo apt-get install mysql-server php5-mysql
```

Ahora establecemos la contraseña del Root (Sustituir “Contraseña”):

```
mysqladmin -u root password Contraseña
```

Ahora tenemos que crear la base de datos de Moodle en MySQL:

```
mysql -u root -p
> CREATE DATABASE moodle DEFAULT CHARACTER SET utf8 COLLATE utf8_unicode_ci;
> GRANT ALL PRIVILEGES ON moodle.* TO moodleuser@localhost
  IDENTIFIED BY 'Contraseña';
> GRANT SELECT,LOCK TABLES on moodle.* TO moodlebackup@localhost
  IDENTIFIED BY 'Contraseña_Backup_Moodle';
> FLUSH PRIVILEGES;
> QUIT
```

2. Instalar Apache.

Escribimos en la consola lo siguiente para descargarlo:

```
sudo apt-get install apache2 libapache2-mod-php5 php5-gd
```

Opcionalmente, por temas de seguridad, podemos instalar:

```
sudo apt-get install libapache2-mod-security php5-ldap php5-odbc
```

Y reiniciamos Apache:

```
sudo /etc/init.d/apache2 restart
```

3. Instalación de utilidades complementarias.

Escribimos en la consola lo siguiente para descargarlas:

```
sudo apt-get install openssh-server unattended-upgrades
sudo apt-get install unzip zip aspell-en aspell-fr aspell-de aspell-es
sudo apt-get install curl php5-curl php5-xmlrpc
sudo apt-get install clamav-base clamav-freshclam clamav
```

4. Instalación del sistema Moodle.

Lo primero es descargar la versión que queremos del siguiente enlace, recomendamos bajar la última versión estable: <http://download.moodle.org/>

Ahora movemos el archivo a:

```
sudo mv /ubicacion/moodle.tgz /var/www
```

Extraemos su contenido:

```
sudo tar xzf moodle.tgz
```

Creamos el siguiente directorio:

```
sudo mkdir /var/moodldata
```

Y cambiamos permisos:

```
sudo chown -R www-data:www-data /var/moodldata
```

Donde www-data es el grupo creado automáticamente en la instalación de Apache.

Para terminar, reiniciamos apache2:

```
sudo service apache2 restart
```

5. Configuración de Moodle.

Entramos en la dirección: <http://localhost/moodle>

Y nos llevará al archivo de `install.php`, con el que procedemos a la instalación:

- a) Elegimos el idioma.
- b) Confirmamos las diferentes rutas.
- c) Seleccionamos el controlador de la base de datos.
- d) Aceptamos los términos y condiciones.
- e) Se realizan las comprobaciones del servidor.
- f) Esperar que se complete la instalación y continuar.
- g) Configuramos la cuenta del administrador y completamos la descripción del sitio.

Y ya tenemos completada la instalación del sistema Moodle y tendremos algo parecido a lo que aparece en la Figura [A.1](#).

Ahora procedemos a describir la instalación y configuración del bloque EDUtwitter en este sistema Moodle:

1. Creación de una cuenta en Twitter.

Primero, crearemos una cuenta en Twitter para asociarla al curso si no lo hemos hecho ya. Ingresamos en <http://www.twitter.com> y rellenamos el formulario de registro que aparece en la Figura [A.2](#).

2. Registrar la aplicación en Twitter.

Para poder acceder a la API de Twitter, necesitamos registrar una aplicación. Para ello ingresamos en la web <http://dev.twitter.com>. Iniciamos sesión en esta página con los datos de nuestra cuenta de Twitter y veremos un recuadro arriba a la derecha como el de la Figura [A.3](#). Pulsamos en “My applications” y luego en “Create a new application”. Rellenamos el formulario que nos aparecerá y nos mostrará una pantalla como la de la Figura [A.4](#). Tenemos que cambiar el “Access level” para que sea de lectura y escritura (Read and write). Apuntamos las claves “Consumer key” y “Consumer secret”. Generamos abajo el access token y apuntamos también el “Access token” y el “Access token secret”. Estas cuatro claves tendremos que ponerlas en el último paso de configuración del bloque.



Figura A.1: Entorno Moodle

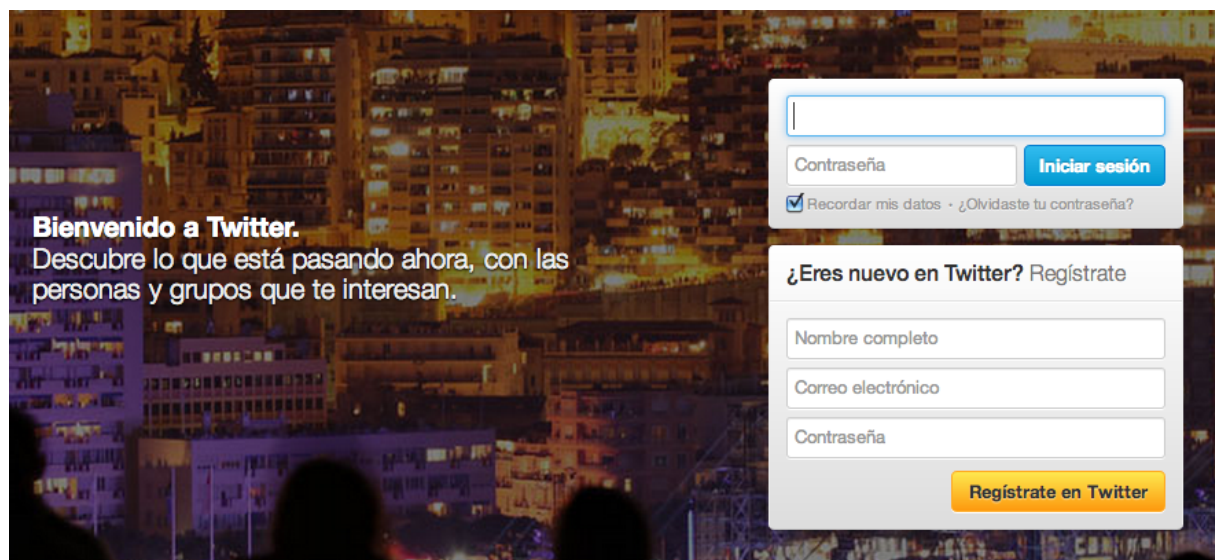


Figura A.2: Página de inicio de Twitter

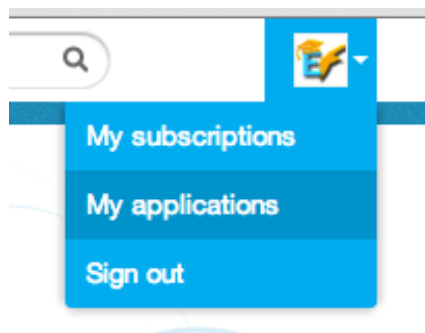


Figura A.3: Seleccionar “My applications”

OAuth settings

Your application's OAuth settings. Keep the "Consumer secret" a secret. This key should never be human-readable in your application.

Access level	Read-only About the application permission model
Consumer key	mc3VlXIPWnB8ADRCq6FsQ
Consumer secret	KcoWRdMoVbf0qF7E9Jwof2Ja3GBKWgA6eIi4X12L5Ck
Request token URL	https://api.twitter.com/oauth/request_token
Authorize URL	https://api.twitter.com/oauth/authorize
Access token URL	https://api.twitter.com/oauth/access_token
Callback URL	None
Sign in with Twitter	No

Your access token

Use the access token string as your "oauth_token" and the access token secret as your "oauth_token_secret" to sign requests with your own Twitter account. Do not share your oauth_token_secret with anyone.

Access token	442816413-Dk5SRMMTrLp3QuhcFSt5fq4aKZu7bEBjcwQQdMb
Access token secret	EHWvKrfnn7Rol2lt4SrWDzJMashKnOxYwSPR1L4YG70
Access level	Read-only

[Recreate my access token](#)

Figura A.4: Datos de la aplicación

3. Copia de la carpeta del plugin en el directorio de Moodle.

Lo único que tenemos que hacer es copiar la carpeta “edutwitter” dentro de la carpeta “blocks” de la raíz de instalación de Moodle. Por ejemplo, “/var/www/moodle/blocks”. Si todo ha ido bien, la próxima vez que iniciemos sesión como administrador en Moodle, aparecerá una pantalla que informa del encuentro de un nuevo bloque. Sólo tendremos que pulsar el botón “Actualizar” para que se complete la instalación.

4. Añadir el bloque EDUtwitter a un curso.

Una vez en el curso con algún rol con privilegios de edición, activar la edición pulsando en el botón de arriba a la derecha. Debajo de los bloques añadidos, tendremos una lista desplegable para añadir un nuevo bloque, en este caso elegimos EDUtwitter. La página se recargará con el bloque añadido y podremos situarlo donde queramos.

5. Configuración de EDUtwitter.

Con la edición aún activa, pulsamos en el botón de configuración del bloque. Nos llevará a un formulario como el de la Figura A.5. En “Nombre” ponemos el nombre de usuario de la cuenta que queremos asociar con el curso. En los siguientes cuatro campos ponemos las claves que nos dieron al registrar la aplicación en Twitter. En “Hashtag” tenemos que poner el hashtag asociado al curso. Podemos generarlo con un botón (genera una cadena con el identificador del curso seguido de una letra aleatoria) y comprobar que no se ha usado ese hashtag en la última semana con otro. Los parámetros hasta ahora establecidos no deberían cambiarse a lo largo del curso. En “Idioma en Twitter” ponemos el idioma en que queremos que aparezcan los tweets de la actividad reciente (de momento inglés o español). Los siguientes cuatro campos configuran los tweets recientes mostrados en el bloque. Por último, elegimos si queremos que se publique la actividad reciente en Twitter y si se cuentan los tweets para generar estadísticas. Guardamos los cambios y el bloque ya está totalmente listo para su uso.

A.5. Pruebas de implantación

Una vez instalado el sistema Moodle y añadido el bloque EDUtwitter configurado a cualquier curso, debemos hacer las siguientes comprobaciones:

1. Abrir el navegador web y escribir la dirección de dominio donde hemos instalado Moodle o, si lo hemos instalado en nuestra máquina de forma local, ir a <http://localhost/moodle>. Tiene que aparecer el inicio del portal de Moodle.
2. Iniciamos sesión con nuestro nombre de usuario y contraseña en Moodle y navegamos hasta el curso donde hemos añadido el bloque. Aparecerá el bloque donde lo hayamos colocado. Si ya hay algún tweet con el hashtag del curso en Twitter, se mostrará dentro de éste.
3. Enviamos un tweet desde Moodle y comprobamos que se haya enviado correctamente en Twitter. Así sabremos que la configuración de los parámetros de la cuenta está bien realizada.
4. Registramos el nombre de usuario en Twitter de la cuenta enlazada con el curso y generamos estadísticas, donde deberá aparecer ese nombre registrado, para comprobar que los accesos a la base de datos son correctos.

Configurando un bloque EDUtwitter

Ajustes de bloque

Nombre	EDUtweets_CUSL6
Consumer key	dzeysOBGROp2KRLgijlxp
Consumer secret	T3SvZaXG0EPJmrf6nNu:
Access token	442816413-QP7Z168Zf1Q
Access token secret	NapUZPcXpgIJ26z6djes7l
Hashtag #	0003B

Idioma en Twitter Español ▾

Número de tweets 5 ▾

Mostrar nombres reales ☒

Mostrar nombres de usuario ☒

Mostrar imágenes de perfil ☒

Publicar en Twitter la actividad reciente ☒

Generar estadísticas ☒

Figura A.5: Configuración de un bloque EDUtwitter

5. Escribimos algún mensaje de prueba en alguno de los foros.
6. Como administradores del sistema, vamos a “Notificaciones” y ejecutamos el script cron.php, no debe dar ningún error. Ahora aparecerá en Twitter el mensaje puesto en el foro. Además, volvemos a generar estadísticas y comprobamos que se haya actualizado el número de tweets del usuario asociado al curso con el tweet que enviamos en uno de los pasos anteriores.

A.6. Procedimientos de operación y nivel de servicio

El administrador del sistema debe ejecutar o programar la ejecución del script cron.php de Moodle un mínimo de una vez cada 24 horas. La publicación de la actividad reciente y el recuento necesario para las estadísticas se realiza solo cada vez que se ejecuta dicho script, al igual que funciona el envío de correos electrónicos de las suscripciones a los foros. Esto asegura el mantenimiento del sistema.

Para realizar copias de seguridad o backups, tenemos dos modos:

- Moodle tiene su función propia para realizar y recuperar copias de seguridad por cada curso. El administrador del curso puede encontrar esta opción dentro de las acciones de “Administración del curso”. La copia de seguridad es totalmente personalizable con lo que se quiere guardar (sólo recursos, mensajes del foro...) y se pueden descargar y recuperar en otro sistema Moodle.
- El administrador del sistema debería realizar periódicamente o automatizar (al menos una vez por semana) una copia de seguridad del sistema completo en una memoria secundaria distinta al original, por si se produjera algún error en ésta. Para esto, hay que realizar una copia de seguridad de la base de datos, de la carpeta moodledata y de los archivos de apache. La copia de seguridad de la carpeta moodledata puede ser incremental para que sea más eficiente. En un sistema linux por ejemplo, podemos hacer todo lo anterior con rsync y mysqldump (si la base de datos es MySQL) programado con crontab.

Apéndice B

Manual de usuario

Las instrucciones de uso del sistema se detallan a continuación.

B.1. Introducción

EDUtwitter es un bloque para la plataforma educativa Moodle, creado para poder aprovechar las características que nos ofrece la red social Twitter para mejorar la comunicación y complementar la evaluación entre los participantes de un curso.

La comunicación entre profesores y entre alumnos y profesores en el aula virtual puede resultar lenta en algunos casos en los que la llegada de información debe ser rápida y eficaz, como el retraso o cambio de aula de una clase. Además, el desarrollo de las comunicaciones móviles ha propiciado el intercambio de mensajes cortos y la interfaz de Moodle no está optimizada para estos dispositivos.

Actualmente el uso de Twitter está muy extendido y se adapta a la inmediatez de las aplicaciones móviles en el bolsillo. EDUtwitter aprovecha esto y lo integra junto con un curso del aula virtual para que las comunicaciones, mensajes, opiniones y demás lleguen al conocimiento de los profesores y estudiantes lo antes posible. Además, recopila el uso que hacen los alumnos de esta red social para hablar del curso y así complementar su evaluación final.

B.2. Características

Las principales características del bloque son:

- Permite enlazar una cuenta de Twitter con un curso de Moodle. También genera y usa un hashtag propio del curso.
- Hace posible enviar tweets a través de Moodle sin necesidad de acceder a Twitter.
- Muestra en la página de Moodle los últimos tweets enviados con el hashtag del curso.
- Registra en una base de datos cada alumno con su nombre de usuario en Twitter.
- Muestra estadísticas con la participación a través de Twitter de los alumnos.
- Envía automáticamente tweets con la actividad reciente del curso.

B.3. Requisitos previos

EDUtwitter no tiene más requisitos que los necesarios para instalar Moodle.

B.3.1. Requisitos del cliente

El único requisito para el cliente es poseer un navegador web. Debería funcionar con cualquier navegador moderno (Internet Explorer versión 6 y anteriores no están soportados).

B.3.2. Requisitos del servidor

Requisitos software

- Un sistema operativo. Cualquiera que soporte el siguiente software. Linux y Windows suelen ser las opciones comunes y las que tienen el mejor soporte.
- Un servidor web. Preferiblemente Apache o ISS.
- PHP. La versión mínima permitida es 5.3.2.
- Una base de datos. MySQL (versión mínima 5.0.25) y PostgreSQL (versión mínima 8.3) son las bases de datos de desarrollo primarias, las más probadas y tienen una documentación y soporte extensos. Oracle (versión mínima 10.2) y MSSQL (versión mínima 9.0) están totalmente soportados, pero la documentación no es tan buena. Se necesitará la extensión PHP apropiada para la base de datos elegida.

Requisitos hardware

- Espacio en disco: 160MB libres (mínimo) más tanto como se necesite para almacenar los recursos y materiales. 5GB es probablemente un mínimo realista.
- Espacio para copias de seguridad: Al menos el mismo espacio que el anterior (remoto preferiblemente) para almacenar una copia de seguridad del sitio.
- Memoria RAM: 256MB (mínimo), 1GB o más sería muy recomendado. La regla general es que Moodle puede soportar de 10 a 20 usuarios concurrentes por cada 1GB de RAM, pero variará dependiendo del hardware específico, de la combinación de software y del tipo de uso. Usuarios concurrentes quiere decir realizando al mismo tiempo peticiones al servidor, no se refiere al número de usuarios que tienen la sesión iniciada.

B.4. Uso del sistema

El uso del bloque EDUtwitter dependerá del rol que tengamos: estudiante o profesor.

B.4.1. Uso como estudiante

Visión del bloque

El alumno podrá ver el bloque tal y como lo haya configurado el profesor. En él verá un número determinado de tweets recientes con el hashtag del curso. En la Figura B.1 se puede ver un ejemplo de lo que vería. Se observa que a pie de bloque hay un enlace que lo llevará a la siguiente funcionalidad.



Figura B.1: Bloque con el rol de estudiante

The screenshot shows a web form titled "Datos de registro". Inside the form, there is a message: "Si envías varios nombres de usuario usando este registro, en la base de datos quedará guardado el último que envíes". Below this, there is a label "Nombre de usuario en Twitter" with a red asterisk, followed by a text input field containing "EDUtweets_CUSL6". At the bottom of the form, there are two buttons: "Guardar cambios" and "Cancelar". Below the form, there is a red text note: "En este formulario hay campos obligatorios*".

Figura B.2: Formulario para enviar el nombre de usuario en Twitter

Enviar nombre de usuario en Twitter

Pulsando en el enlace al pie del bloque “Enviar usuario en Twitter”, se dirigirá al usuario a un formulario como el de la Figura B.2. Sólo tiene que introducir su nombre de usuario en Twitter (Sin la @ inicial identificativa propia de la red social) en el cuadro de texto blanco y pulsar en el botón “Guardar cambios”. Si comete algún error en la escritura, sólo tiene que volver a enviar su nombre de usuario de forma correcta. En la base de datos se almacenará sólo el último nombre de usuario enviado por cada alumno.

B.4.2. Uso como profesor

Además de los usos como estudiante, el profesor puede realizar las siguientes funcionalidades.

Visión del bloque

La visión del bloque como profesor es distinta para poder acceder a nuevas funcionalidades. El bloque se verá como aparece en la Figura B.3.

Enviar Tweet

Pulsando en el enlace al pie de bloque “Enviar Tweet”, se dirigirá al usuario a un formulario como el de la Figura B.4. El formulario permite incluir automáticamente al inicio del tweet el hashtag propio del curso y escribir un tweet de hasta 250 caracteres. Twitter sólo permite tweets de hasta 140 caracteres, por lo que un tweet enviado desde este formulario con más de esa cantidad de caracteres será dividido en el número de tweets necesarios de hasta 140 caracteres para poder ser publicados. Sólo hay que escribir en el recuadro en blanco y pulsar en el botón “Guardar cambios” para que se envíe el tweet desde la cuenta asociada al bloque.

Generar estadísticas

Pulsando en el enlace al pie de bloque “Estadísticas”, se dirigirá al usuario a un formulario como el de la Figura B.5. Antes de pulsar el botón “Guardar cambios” para generar las estadísticas, puede marcar la casilla “Reiniciar estadísticas”. Si se marca, una vez generada las estadísticas se pondrán todos los contadores a 0 de nuevo. Una vez pulsado el botón, aparecerá una tabla como la de la Figura B.6. Aquí aparecen los alumnos del curso que han registrado su nombre de usuario en Twitter ordenados alfabéticamente por sus apellidos para facilitar la evaluación, el



Figura B.3: Bloque con el rol de profesor

Datos de publicación

Incluir hashtag automáticamente

Tweet*

En este formulario hay campos obligatorios*

Figura B.4: Formulario para enviar un tweet desde Moodle

Estadísticas

Pulsa el botón "Guardar cambios" y aparecerá una tabla con las estadísticas de los tweets con el hashtag del curso enviados por cada alumno. Si marcas la casilla "Reiniciar estadísticas" antes de hacerlo, una vez que muestre la tabla reiniciará los contadores de tweets de cada alumno.

Reiniciar ☐ estadísticas

Guardar cambios

Cancelar

Figura B.5: Formulario para generar estadísticas

Estadísticas				
Apellidos	Nombre	Nombre de usuario en Twitter	Número de tweets	Porcentaje del total
García	José	@JoseM	6	22.22%
López	Jesús	@JesussTT	5	18.52%
Martínez	Juan	@JuanM	3	11.11%
Pérez	Luis	@LuisPtwitter	9	33.33%
Romero	David	@_Hocicos_	4	14.81%

Figura B.6: Tabla generada con las estadísticas por alumno

número de tweets que ha escrito con el hashtag del curso y el porcentaje sobre el número total de tweets escrito por todos los que aparecen en la tabla.

Apéndice C

Manual del desarrollador

Las instrucciones de instalación y explotación del sistema se detallan a continuación.

C.1. Introducción

EDUtwitter es un bloque para la plataforma educativa Moodle, creado para poder aprovechar las características que nos ofrece la red social Twitter para mejorar la comunicación y complementar la evaluación entre los participantes de un curso.

La comunicación entre profesores y entre alumnos y profesores en el aula virtual puede resultar lenta en algunos casos en los que la llegada de información debe ser rápida y eficaz, como el retraso o cambio de aula de una clase. Además, el desarrollo de las comunicaciones móviles ha propiciado el intercambio de mensajes cortos y la interfaz de Moodle no está optimizada para estos dispositivos.

Actualmente el uso de Twitter está muy extendido y se adapta a la inmediatez de las aplicaciones móviles en el bolsillo. EDUtwitter aprovecha esto y lo integra junto con un curso del aula virtual para que las comunicaciones, mensajes, opiniones y demás lleguen al conocimiento de los profesores y estudiantes lo antes posible. Además, recopila el uso que hacen los alumnos de esta red social para hablar del curso y así complementar su evaluación final.

C.2. Uso de EDUtwitter por parte de otros módulos

Un desarrollador de otro módulo o actividad de Moodle podría buscar que EDUtwitter publicara también la actividad reciente de su plugin.

Esto es bastante sencillo de realizar, sigue un patrón muy parecido que el bloque de actividad reciente. Para el bloque de actividad reciente, en el archivo `lib.php` de su módulo el desarrollador debe implementar la función `[nombre_del_modulo]_print_recent_activity`. En el caso de EDUtwitter hay que tener algunas cosas más en cuenta, pero es bastante sencillo. Los pasos a realizar serían los siguientes:

1. Copiar los archivos `TwitterAPIExchange.php` y `split_tweet.php` de la carpeta de `edutwitter` a la carpeta de su módulo e incluirlos en el fichero `lib.php`. Estos archivos son necesarios para la autenticación en Twitter y para dividir los tweets largos de la actividad reciente respectivamente.
2. En el archivo `lib.php` implementar la función:

```
[nombre_del_modulo]_print_recent_activity_twitter($course, $viewfullnames,
$timestamp, $block)
```

Donde \$course es el objeto del curso, \$viewfullnames los privilegios para acceder a algunos datos, \$timestamp el tiempo desde cuando han de mirarse las novedades y \$block el objeto del bloque EDUtwitter.

Al principio de la función, incluir el siguiente bloque de código para hacer la llamada a la API:

```
/**
 * Variables needed for the Twitter API
 */
$consumerkey=$block->config->consumerkey;
$consumersecret=$block->config->consumersecret;
$oauthtoken=$block->config->oauthtoken;
$oauthsecret=$block->config->oauthsecret;
$language=$block->config->language;
if($block->config->search_term[0] == '#'){
    $search_term=$block->config->search_term;
} else {
    $search_term='#'.$block->config->search_term;
}

$settings = array('oauth_access_token' => $oauthtoken,
'oauth_access_token_secret' => $oauthsecret,
'consumer_key' => $consumerkey, 'consumer_secret' => $consumersecret);
$url = 'https://api.twitter.com/1.1/statuses/update.json';
$requestMethod = 'POST';
```

Hay que crear una variable \$contenido que guarde la cadena enviada a través de un tweet. Debe hacerse de la siguiente forma para conservar el bilingüismo:

```
if($language == '1'){
    $contenido=$search_term.' '. 'contenido en español con el
    hashtag delante';
} else {
    $contenido=$search_term.' '. 'contenido en inglés con el
    hashtag delante';
}
```

Por último, el código para enviar el tweet es el siguiente:

```
$postfields = array('status' => $contenido);
$tweet = new TwitterAPIExchange($settings);
$response = $tweet->buildOauth($url, $requestMethod)
->setPostfields($postfields)
```

```

        ->performRequest();
$salida = json_decode($response);
if(!empty($salida->errors)){
    foreach($salida->errors as $error){
        if($error->code == '186'){
            split_tweet($tweet,$contenido);
        }
    }
}
}

```

Llamamos a `split_tweet` para dividir el tweet en varios si el contenido excede los 140 caracteres. Con esto quedaría terminada la función y la actividad reciente del módulo se publicaría en Twitter haciendo uso del bloque EDUtwitter.

C.3. Ampliar EDUtwitter

C.3.1. Añadir nuevo idioma

EDUtwitter es un bloque un poco distinto del resto a la hora de añadir un nuevo idioma. Ya que no sólo hay que tener en cuenta el idioma de la interfaz en Moodle, sino también el idioma en el que aparecen los tweets de la actividad reciente en Twitter.

Nuevo idioma en la interfaz de Moodle

Para añadir un nuevo idioma del bloque en Moodle, es exactamente igual que el resto de los bloques. En la carpeta “lang” podemos encontrar dos más, “es” (idioma español) y “en” (idioma inglés). Sólo tenemos que añadir una nueva carpeta con las iniciales del nuevo idioma y copiar dentro el archivo “block_edutwitter.php” de una de las carpetas anteriores, el idioma del que prefiramos traducir. El archivo es un diccionario, sólo hay que traducir cada cadena que aparece y ya tenemos un nuevo idioma en la interfaz del bloque.

Nuevo idioma en la publicación de mensajes en Twitter

Este proceso es más complicado, ya que hay que modificar dos de los archivos de EDUtwitter y todas las funciones de actividad reciente que hagan uso de él.

- `edit_form.php`.

Cuando añadimos el elemento “config_language”, tenemos que poner en el array el nuevo idioma para que podamos elegirlo en la configuración del bloque.

- `recent_activity.php`

Cada vez que se inserta en la variable “\$contenido” un tweet, comprobarás que hay una estructura “if” que elige el lenguaje. Sólo hay que modificarla para añadir una tercera opción.

- Actividad reciente de EDUtwitter en otros bloques.

Recorrer el resto de módulos que tengamos instalados que hayan implementado una función “[nombre_módulo]_print_recent_activity_twitter” y modificarla de igual forma que el archivo anterior.

C.3.2. Añadir nuevas funciones

La forma de añadir nuevas funciones depende de lo que quiera hacer el desarrollador, aquí dejamos una descripción de lo que contiene cada archivo para que sirva de referencia:

- Carpeta “lang”
Carpeta que contiene los diccionarios de los distintos idiomas en los que está la interfaz del bloque en Moodle.
- Carpeta “db”
Carpeta con la instalación de la base de datos necesaria realizada con el editor XMLDB de Moodle.
- block_edutwitter.php
Archivo principal del bloque. Contiene la clase block_edutwitter, las instrucciones de lo que debe mostrarse en el bloque y la actividad que se debe realizar con la ejecución del script cron.php.
- edit_form.php
Contiene los elementos que se añaden a los predeterminados en la configuración del bloque.
- version.php
Contiene información relativa a la versión necesaria de Moodle para instalar el bloque, la versión de EDUtwitter y la frecuencia con la que se ejecuta dentro del script cron.php.
- TwitterAPIExchange.php
Librería para realizar peticiones a la API 1.1 de Twitter.
- submit_form.php y view_submit.php
Contienen los elementos del formulario para enviar el nombre de usuario en Twitter y la forma de procesar esos datos respectivamente.
- sendtweet_form.php y view_sendtweet.php
Contienen los elementos del formulario para enviar un tweet y la forma en que se procesa el envío de ese tweet respectivamente.
- statistics_form.php y view_statistics.php
Contienen los elementos del formulario para generar estadísticas y la forma en que se crea la tabla y se muestran los datos respectivamente.
- searchtwitter.php y styles.css
Contienen la función para buscar y mostrar los tweets recientes con el hashtag del curso y los estilos css en que se muestran respectivamente.
- split_tweet.php
Contiene la función que hace posible dividir un contenido de más de 140 caracteres en varios tweets y publicarlos.

- `count_tweets.php`

Contiene la función que cuenta los tweets con el hashtag del curso publicados por los alumnos para la generación de estadísticas.

Bibliografía

- [1] Moodle (2013). Página oficial Moodle. <http://www.moodle.org>.
- [2] Twitter (2013). Página oficial Twitter. <http://www.twitter.com>.
- [3] CCOO (2010). Tablas salariales 2010 IV Convenio Colectivo, <http://www.uca.es/sindicato/ccoo/documentos/tabla-salarial-pas-laboral-2010.pdf>.
- [4] The Apache Software Foundation (2013). Página web oficial. <http://www.apache.org/>
- [5] GNU Emacs (2013). Página web oficial. <http://www.gnu.org/software/emacs/>
- [6] Forja RedIRIS (2013). Página con los repositorios. <https://forja.rediris.es/>
- [7] Twitter-API-PHP (2013). Documentación oficial de la librería. <https://github.com/J7mbo/twitter-api-php>

Otras referencias:

- [8] Ignacio J. Sánchez Rojo (2010). *Plataforma educativa Moodle. Administración y gestión*. ISBN: 978-8478979455.
- [9] Kevin Makice (2009). *Twitter Api: Up and Running*. ISBN: 978-0596154615.
- [10] William Rice (2011). *Moodle 2.0 E-Learning Course Development*. ISBN: 978-1849515269.
- [11] Luke Welling, Laura Thomson (2009). *Desarrollo Web con PHP y MySQL*. ISBN: 978-8441525535.
- [12] Forja RedIRIS EDUtwitter (2013). Forja oficial de EDUtwitter. <https://forja.rediris.es/projects/cusl6-edutweets/>
- [13] David Romero Santos (2013). Blog oficial de EDUtwitter. <http://edu-twitter.blogspot.com>
- [14] Plugins Moodle (2013). Repositorio de bloques de Moodle. <https://moodle.org/plugins/>
- [15] Documentación Moodle (2013). Documentación oficial para desarrolladores. http://docs.moodle.org/dev/Main_Page
- [16] Documentación Twitter (2013). Documentación sobre la Rest API de Twitter. <https://dev.twitter.com/docs/api>

- [17] Twitter Search Block (2013). Bloque de búsqueda en Twitter. https://moodle.org/plugins/view.php?plugin=block_twitter_search
- [18] EduTwitter (2013). Web sobre el microblogging en la educación. <http://edutwitter.wikispaces.com/>

GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.

`<http://fsf.org/>`

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “**Document**”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “**you**”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “**Modified Version**” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “**Secondary Section**” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary

Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “**Invariant Sections**” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “**Cover Texts**” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “**Transparent**” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “**Opaque**”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “**Title Page**” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

The “**publisher**” means any person or entity that distributes copies of the Document to the public.

A section “**Entitled XYZ**” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “**Acknowledgements**”, “**Dedications**”, “**Endorsements**”, or “**History**”.) To “**Preserve the Title**” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of

Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements”.

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate”

if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

“Massive Multiauthor Collaboration Site” (or “MMC Site”) means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A “Massive Multiauthor Collaboration” (or “MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

“CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

“Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is “eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright © YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with ... Texts.” line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.